

I NFORMATYKA
K OMPUTERY
S YSTEMY



CENA — 80 ZŁ

DODATEK „ŻOŁNIERZA WOLNOŚCI” NR 10/1987 ISSN 0860—2794

29.XI.1987r.

Gra o Jutro



TMMT

Każdy nadesłany do redakcji list jest uważnie przeczytany. Staramy się w redagowaniu pisma wykorzystywać pomysły naszych Czytelników. Są listy, na które odpowiadamy również listami, są także takie, na które odpowiedź jest artykuł czy program zamieszczony na naszych łamach.

Czytelnicy zawsze towarzyszyli powstawaniu każdego numeru „IKS-a”. Z waszych listów, a ostatnio także z nadesłanych ankiet wiemy, czym powinniśmy się zajmować. Dlatego też na naszych łamach przeważać będą informacje na temat możliwości posiadanego przez was sprzętu. Oczywiście pisać nadal będziemy o Atari, Commodore, Spectrum, ale także o Amstradzie, którego liczba użytkowników stale wzrasta i IBM czyli olbrzymiej już rodzinie komputerów od lat wyznaczających światowy standard. Zatem posiadacze osmiobitowych mikrokomputerów nadal będą mogli znaleźć u nas informacje o tym, co jeszcze potrafią ich komputery, a o czym fabryczne instrukcje często milczą. Czasami krótki, zawierający kilkanaście instrukcji program odkryć może nieznane dotąd walory np. starego Spectrum.

To oczywiście nie wszystko. Już od dzisiejszego numeru drukować będziemy co miesiąc angielsko-polski słownik informatyczny. Mamy nadzieję, że ułatwi on studiowanie trudnej obcej literatury fachowej. Na łamy „IKS-a” trafiła również fantastyka, której autorami są nasi Czytelnicy — publikujemy bowiem opowiadania nagrodzone w naszym konkursie.

Jeśli chcecie, aby „IKS” był naprawdę waszą gazetą piszcie do nas. Zawartość „IKS-a” zależy od Czytelników.

W NUMERZE:

Czy będziemy mieli polski mikrokomputer?	— str. 4
W szponach Atari (7)	— str. 5—8
Przystawka magnetofonowa do Atari	— str. 11—13
Procedury Rom SPECTRUM (1)	— str. 14—15
Opowiadanie SF „Final”	— str. 20
Basic (9)	— str. 21—25
Badanie kierowców	— str. 25—27
Emulator Basic 1.1 dla CPC 464 (2)	— str. 27
Informatyczny słownik angielsko-polski	— str. 29—30

Ogólnopolska Federacja Klubów Komputerowych Młodych Mistrzów Techniki ZSMP powstała w styczniu 1986 roku. Obecnie w jej skład wchodzi 264 kluby, wśród których najwięcej jest placówek działających przy zakładach pracy i szkołach. Podstawowym celem Federacji, celem wspólnym dla wszystkich jej ogniw jest szeroko rozumiana edukacja informatyczna młodzieży.

Oczywiście klub to nie szkoła. Wprawdzie uczestnicy zajęć — w większości zresztą uczniowie, zdobywają wiedzę, jednak proces ten ma niewiele wspólnego z lekcjami. Porównać go można raczej do nauki własnej, do samokształcenia, którego kierunki i intensywność zależą od indywidualnych zainteresowań konkretnego człowieka. Zadaniem klubu jest więc stworzenie optymalnych warunków do rozwoju pasji jego członków. Kto ma styczność z jakąkolwiek placówką KK MMT wie jak trudne jest to zadanie.

Oczekiwania fanów informatyki, potrzeby które zgłaszają w dziedzinie sprzętu (jego ilości czy jakości), literatury, wydawnictw, dostępu do oprogramowania, niejednokrotnie przekraczają możliwości organizacyjne i finansowe klubu i jego sponsorów. Jednocześnie wiele problemów łatwiej pokonać, wiele spraw szybciej i taniej można załatwić występując grupowo. Stara to zasada i wielokrotnie już potwierdzona w praktyce. Ona też przesądza o sensowności istnienia Federacji KK MMT, powołanej przecież przez organizatorów i działaczy klubów egzystujących wcześniej w pojedynkę, zdanych tylko na własne siły przy pokonywaniu wspólnych przeciwności dla wszystkich problemów.

Korzystając z gościnnych łamów „IKS-a” będziemy, od tego numeru na bieżąco, informowali o naszych propozycjach i przedsięwzięciach. Sądzę, że będą to sprawy w równej mierze interesujące dla klubów pracujących w formule KK MMT, niż związanych z Federacją, jak też dla indywidualnych posiadaczy mk. Chcemy jednocześnie zaproszować wszystkim chętnym współdziałanie. Więcej — liczymy bardzo na głosy oceniające poszczególne elementy naszej oferty, na rady, opinie i podpowiedzi, na podzielenie się z nami naszymi przemyśleniami i doświadczeniami. Im więcej będzie zabierających głos, im szersze kręgi obejmie rozmowa o naszych sprawach, tym bardziej prawdopodobne jest wypracowanie najlepszych, najbardziej efektywnych metod działania.

Czekając na wasze listy już teraz proponujemy:

1. Pomoc prawną i organizacyjną przy powoływaniu klubów komputerowych.

2. Nieodpłatne pośrednictwo przy zakupach sprzętu komputerowego dla klubów bądź ich sponsorów. Przykładowe ceny: Amstrad CPC 6128 z monitorem

z zielonym — 520 tys., IBM PC/XT (RAM 640, monitor firmowy, karta Hercules, CGA, 1x360 kB — dysk firmy NEC) — 1280 tys. zł, hard-dysk 20 MB — 1200 tys. — gwarancja — od 12 do 18 miesięcy. Upusty, które uzyskaliśmy wynoszą od 10 do 50 % od cen rynkowych.

3. Legitymacje, znaczki, plakiety, plakaty itp. elementy propagandowe (dla placówek KK MMT nieodpłatnie bądź w cenie równej kosztom produkcji).

4. Wydawnictwa serii „Mikrokomputerowe ABC” (nieodpłatnie dla placówek KK MMT).

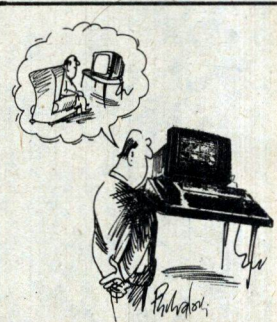
Jednocześnie jeszcze w tym roku będziemy kompletowali i rozsyłali (również nieodpłatnie), klubom zrzeszonym w Federacji, pakiety książek, wydawnictw oraz programów uznanych za najlepsze z dostępnych w kraju. Listę zawierającą konkretne tytuły opublikujemy.

A może, ...może zechcecie nam podpowiedzieć, jakie naszym zdaniem książki i które programy (narzędziowe, użytkowe, dydaktyczne i zabawowe) powinny znaleźć się w takim pakiecie? Jeżeli tak, kontaktujcie się z Sekretariatem Rady Ogólnopolskiej Federacji Klubów Komputerowych Młodych Mistrzów Techniki, Warszawa, ul. Smólna 40 pok. 414 lub dzwonić tel. 26-54-01 wew. 362 lub 232.

Jeżeli chcecie dowiedzieć się szczegółów czytajcie „IKS-a”, słuchajcie też kolejnych emisji „Radiokomputera”, który będziemy dla was współredagować. Dla posiadaczy mk. kompatybilnych z ZX Spectrum wiadomościom dodatkowa — w każdy czwarty piątek miesiąca „Radiokomputer” emitował będzie „Komputerowy Biuletyn Informatyczny” w postaci programów przysyłanych przez popularnego „Spekturusa”.

I na razie to byłoby już wszystko. Pozdrawiam, zapraszam do współpracy.

Krzysztof KRUPA



— Wolę tradycyjną telewizję.

Przystawka magnetofonowa dla Atari

Jedną z najpopularniejszych pamięci masowych, stosowanych w mikrokomputerach domowych, jest magnetofon kasetowy. Komputery Atari 130 XE, 800 XL wykorzystują w tym celu specjalizowany magnetofon (np. XC 12, XC 11). Zakup takiej pamięci związanej jest ze stosunkowo dużym wydatkiem.

W tym artykule znajdziecie opis przystawki umożliwiającej wykorzystanie dowolnego magnetofonu kasetowego (mono lub stereo) do Atari. Koszt przystawki wynosi ok. 2+3 tys. złotych.

Zasada działania

Zapis informacji (programów) na taśmie magnetofonowej realizowany przez Atari oparty jest na zasadzie modulacji częstotliwości. Oznacza to, że jedynie logicznej, odpowiada sygnał o częstotliwości 5,1 kHz, a zero logicznie przebieg o częstotliwości 3,9 kHz. Ciąg bitów, który ma być zapisany w kasecie, zamieniany jest w komputerze na sygnały o tych częstotliwościach. Zapis dokonywany jest z szybkością 600 bitów/sekundę.

Zapisanych w ten sposób informacji komputer nie potrafi samodzielnie odczytać. Potrzebny jest układ, który sygnały o częstotliwościach 5,1 kHz i 3,9 kHz (zapisane na taśmie magnetofonowej) zamieni ponownie na sygnały zrozumiałe dla komputera. Przystawka dokonuje tej zamiany i jednocześnie zapewnia, że poziomy napięć tych sygnałów odpowiadają poziomom napięć TTL, tzn. wysokiemu poziomowi napięcia (+5 V) odpowiada jedynka logiczna, a zero logicznie odpowiada niski poziom napięcia (0V).

Schemat blokowy przystawki przedstawia rys. 1. Przystawka połączona jest z komputerem za pośrednictwem gniazda łącza szeregowego. Z magnetofonem przystawka połączona jest typową 5-stykową wtyczką magnetofonową. Zasilana jest napięciem +5 V z komputera.

W czasie wczytywania programu (magnetofon włączony na odtwarzanie) sygnał m.c.z. magnetofonu doprowadzony jest do wzmacniacza m.c.z. (wzmacniacz operacyjny A). Wzmocniony sygnał podawany jest na aktywne filtry pasmowoprzepustowe B i C. Filtr B nastrojeny jest na częstotliwość 3,9 kHz, a filtr C na 5,1 kHz. Gdy na wejściu filtra B pojawi się przebieg o częstotliwości 3,9 kHz, to będzie on wzmożony. Jeżeli na wejściu tego filtra pojawi się przebieg o innej częstotliwości (np. 5,1 kHz) to filtr nie będzie tego sygnału wzmacniał.

Podobnie pracuje filtr C. Ustawiony jest jednak na inną częstotliwość. Jest to częstotliwość 5,1 kHz. Wyjścia filtrów dołączone są do komputera napięciem zbudowanego na wzmacniaczu D. Ta część przystawki jest odpowiedzialna za właściwe dekodowanie informacji. Jeżeli na taśmie pojawi się przebieg o częstotliwości 5,1 kHz („1” logiczna), poprzez magnetofon i wzmacniacz A

```

=AP+J2:FL=J1:GOTO J9
820 AS(J18,J18)="(A*(J18+J1))=S
TR$(PEEK(AP+J1)):A$(22))",V":AP
=AP+J2:FL=J1:GOTO J9
850 AS(J18)="(A*(J18+J1))=STR$(
PEEK(AP+J1)+PEEK(AP+J2)+256):A$(
24)="":AP=AP+J3:FL=J2:GOTO J9
880 FL=J3:AP=AP+J1
900 RESTORE
910 IF AP>AK+J1 THEN GOTO 92
915 A=PEEK(AP-FL)
917 IF A=J0 AND FL<J3 THEN A$(35
,35)="0":A$(49,50)="00":GOTO 925
920 IF FL<J3 THEN A$(39-INT(CLOG
(A)),39)=STR$(A):C=J2:GOSUB 335:
C=J1:GOSUB 330:A$(49,50)=ROZ$(J1
,J2)
925 A=PEEK(AP-J1)
927 IF A=J0 AND FL=J2 THEN A$(39
,39)="0":A$(52,53)="00":GOTO 980
930 IF FL=J2 THEN A$(39-INT(CLOG
(A)),39)=STR$(A):C=J2:GOSUB 335:
C=J1:GOSUB 330:A$(52,53)=ROZ$(J1
,J2)
950 A$(56)=" "
955 IF BRKF=J1 THEN ? ? CHR$(12
7):CHR$(30):"BRK"? :BRKF=J0:IF
DRU THEN LPRINT :LPRINT " ", "

```

```

1102 LPRINT " adres rozk
aż w jezuku kod dzies.
kod hex."
1104 LPRINT "dzies.hex. a
semblera"

```

```

1105 LPRINT "-----
"
960 ? A$(J1,5):A$(13,24):A$(28,3
9):A$(45,53)
980 IF DRU THEN LPRINT A$(J1,11)
" ",A$(12,28):" ",A$(2
9)
990 GOTO 100
995 REM
996 REM
997 REM MODUL WCZYTYWANIA PRO-
998 REM GRAMU DO PRACIECI
999 REM
1000 ? "Chcesz wprowadzić progra
m do pamięci"? "który będzie di
sassemblowany ? (T/N)"
1002 INPUT SPEC$:IF SPEC$(1,1)="
N" THEN 1000:IF SPEC$(1,1)<"T"
THEN GOTO 1002
1008 ? ? "Wolny obszar pamięci"
? "od "?:MP=PEEK(144)+256:KPEEK
(145):? MP:?" do "
1010 MP1=PEEK(741)+256:KPEEK(742)
: ? MP1
1020 ? "Podaj adres bufora "?:IN
PUT ABUF
1030 IF ABUF<MP1 OR ABUF>MP THEN
? "Zły adres " :GOTO 1020
1040 ? "Podaj nazwę zbioru "?:IN
PUT SPEC$
1050 GS=0:IF SPEC$(1,1)="C" THEN
GS=128
1060 TRAP 1040:CLOSE #2:OPEN #2,
4,GS,SPEC$:BH=INT(ABUF/256):BL=A
BUF-(BH*256):POKE 866,?:POKE 872
:0
1065 POKE 873,PEEK(742)-PEEK(145
):POKE 868,BL:POKE 863,BH:X=USR(
ADR(C10$))
1070 TRAP 40000:LZB=PEEK(873):#25
6=PEEK(872):? "Zaladowano " :LZB
: ? " bajtow"

```

* SUMA KONTROLNA / ETYKIETA *

HQ	11	UR	21	J1	31	UT	4
HU	51	YH	101	GK	111	VE	12
OJ	141	LN	151	EH	161	LR	25
JV	271	WQ	281	VH	301	YO	32
OS	341	MI	401	XW	421	LB	45
CB	471	GR	491	XV	511	ZN	53
NL	551	GO	571	IH	591	MU	61
PY	631	UZ	651	YU	671	KV	69
OG	711	EG	731	AC	751	CJ	77
EB	791	IP	811	BI	831	KG	85
RU	871	SL	891	GJ	901	CG	91
IU	921	YN	931	ST	941	ST	95
PX	961	JE	971	ZV	991	ZV	100
GW	1101	XY	3901	FX	3951	HI	397
WL	4001	IX	4011	LR	4021	OS	405
WS	4101	GT	4201	ZU	4301	CX	435
IL	4381	BG	4401	KM	4501	BX	460
QO	4701	AO	4901	YS	5201	MV	550
MK	5801	NO	6101	OY	6401	NP	670
PJ	7001	PH	7301	NP	7601	OH	770
BV	7901	FO	8201	RL	8501	VY	880
RC	9001	FT	9101	AF	9151	UK	917
EM	9201	LZ	9251	BY	9271	EQ	930
SX	9501	WI	9551	XZ	9601	OR	980
MG	9901	VL	9951	CA	9961	VS	997
UA	9981	VX	9991	XU	10001	TZ	1002
QF	10081	BS	10101	WD	10201	UJ	1030
FR	10401	EJ	10501	RD	10601	WD	1055
TS	10701	PR	10801	XY	10901	UD	1091
JR	10921	UL	10931	YO	10941	SM	1100
WI	11021	LQ	11041	TK	11061	OY	1110

podawany jest on na filtry B i C. W tym wypadku zadziała jedynie filtr C. Na jego wyjściu pojawi się wzmacniony sygnał, który steruje wejściem odwracającym komparatora. Napięcie na wejście nieodwracające w tym momencie jest niższe. W związku z tym na wyjściu komputera będzie niski poziom napięcia (prawie 0 V). Napięciem tym sterowany jest końcowy stopień odwracający fazę napięcia. Na wyjściu tego stopnia (czyli jednocześnie na wejście komputera) będzie wysoki poziom napięcia (prawie +5 V), czyli poziom odpowiadający jednemu logicznemu.

Kiedy na taśmie pojawi się przebieg o częstotliwości 3,9 kHz, to zadziała filtr B i na wyjściu komputera będzie wysoki poziom napięcia (napięcie na wyjście „+” wyższe od napięcia na wejście „-”). Po odwróceniu fazy w układzie końcowym na wejściu komputera będzie niski poziom napięcia, odpowiadający zeru logicznemu.

W czasie nagrywania programów na kasecie (magnetofon w trybie zapis) sygnał z komputera podawany jest na dzielnik napięcia. Tak zmniejszony podawany jest na wejście magnetofonu i zapisany na taśmie.

Opis konstrukcji

Schemat ideowy układu zamieszczonego na rysunku 2. Do budowy wykorzystano jeden układ scalony typu LM 324. Jest to powszechny wzmacniacz operacyjny. Cena tego układu na warszawskiej giełdzie wynosi od 400 do 600 złotych. Należy w tym miejscu podkreślić, że ten układ nie może być zastąpiony polskimi wzmacniaczami operacyjnymi (pojedynczymi) typu ULY 7741, ULY 7701 ze względu na to, że nie pracują one poprawnie przy zasilaniu napięciem +5 V. W opisywanej konstrukcji sygnały z filtrów B i C podawane są na komparator poprzez mostek prostowniczy D3 + D6. Rola komparatora odgrywa jeden ze wzmacniaczy operacyjnych. Sygnał z komparatora steruje, poprzez filtr RC, tranzystorem T1. Przebiegi z kolektora tranzystora podawane są do wejścia „IN” komputera.

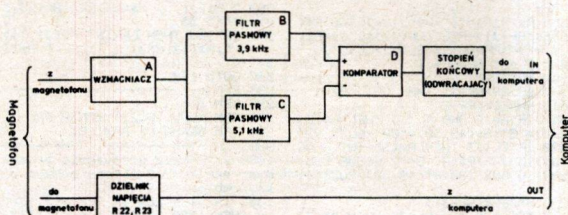
Przystawkę zmontowano na jednostronnie drukowanej płytce o wymiarach 10 x 6 cm (rys. 3). Przystawka łączona jest z komputerem wtyczką dołączoną do gniazda interfejsu szeregowego.

Jeśli nie mamy możliwości zakupienia firmowej wtyczki, to wykonać ją można we własnym zakresie według rysunku 5. Jako elementów stykowych użyto styków z gniazda szufladowego Eltra typu 881025. Po wykonaniu obu detali łącza i uzbrojeniu części A w styki sklejamy je klejem (np. distalem). Połączenia. Wtyczkę z przystawką łączymy (z w skrętkami (przewód sygnałowy okryty przewodem masy).

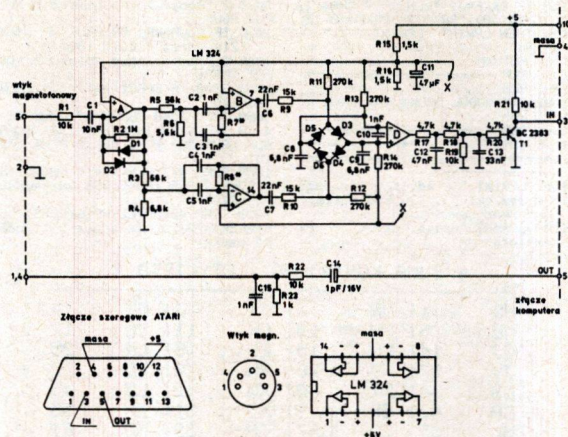
Wtyk magnetofonowy łączymy z układem dwuzwojowym przewodem ekranowym.

Uruchomienie

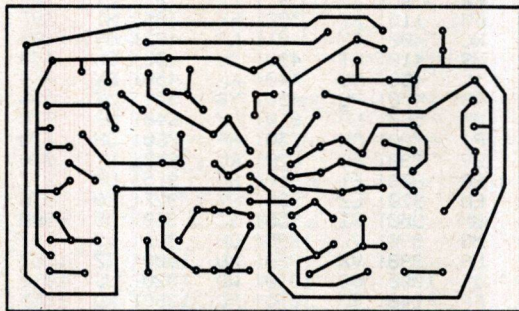
Po zlutowaniu wszystkich elementów i sprawdzeniu poprawności montażu możemy przystąpić do strojenia. Przed tym należy zamiast rezystorów R7 i R8 wlotować potencjometry montażowe 470 kΩ. Przystawkę zasilamy napięciem +5 V (nie z komputera), łączymy z magnetofonem, do którego wkładamy kasetę z nagrany programem dla Atari. Do zestrojenia przystawki musimy użyć oscyloskopu (jest w pracowni fizycznej każdej szkoły średniej), w którym podślawę czasu ustawiamy na 5 + 10 ms. Włączamy magnetofon na odta-



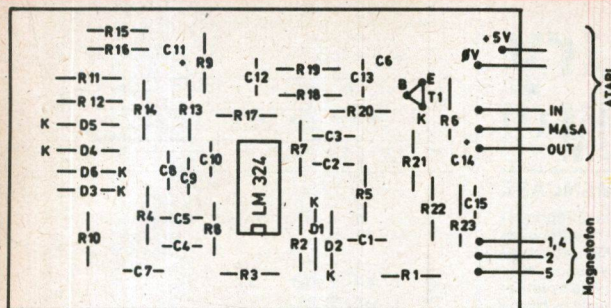
Rys. 1 Schemat blokowy przystawki



Rys. 2 Schemat ideowy przystawki

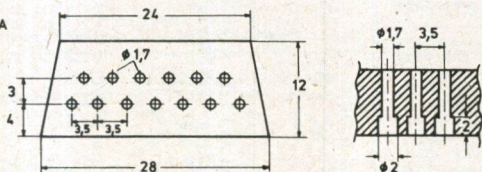


Rys. 3 Płytkę montażową — układ strony druku



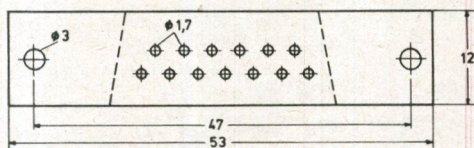
Rys. 4 Rozmieszczenie elementów

DETAL A



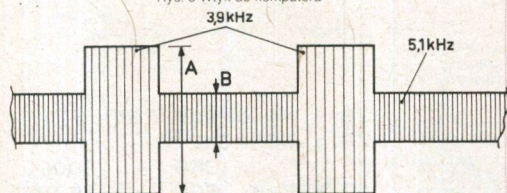
Materiał: tekstolit gr. 8 mm

DETAL B

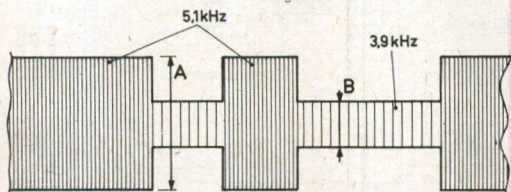


Materiał: tekstolit gr. 1,5 + 2 mm

Rys. 5 Wytyk do komputera



Rys. 6 Oscylogram sygnału na końcówce nr 7 układu scalonego



Rys. 7 Oscylogram sygnału na końcówce nr 14 układu scalonego

zanie i kolejno dołączamy oscyloskop do końcówek 7 i 14 układu scalonego. Kręćąc potencjometrem starajmy się uzyskać na oscyloskopie przebiegi podobne do oscylogramów z rys. 6 i 7. W danym momencie stroimy jeden filtr. Podczas strojenia należy uzyskać jak największą amplitudę A przy jak najmniejszej jednoczesnie amplitudzie B. W najgorszym wypadku amplituda A powinna być przynajmniej dwa razy większa od amplitudy B.

Od tej regulacji zależy poprawia praca przystawką. Po zestrojeniu filtrów mierzymy rezystancje potencjometrów i zamiast nich wylutowujemy rezystory. Możemy sprawdzić jeszcze przebieg na kolektorze tranzystora. W czasie trwania sygnału „pilota” rozpoczynającego program (sygnał o częstotliwości 5 : kHz) na kolektorze powinno być napięcie wyższe od +3.5 V. Po pojawieniu się treści programu na kolektorze powinniśmy obserwować przebieg prostokątny, zmieniający się w takt zapisanej na taśmie informacji.

W czasie użytkowania musimy przestrzegać następujących zasad:

- w pierwszej kolejności łączymy przystawkę z komputerem i dopiero potem włączamy zasilanie komputera.
- w czasie pracy komputera nie wolno odłączać wtyczki przystawki z komputera.
- przystawkę możemy łączyć z magnetofonem w dowolnej chwili.
- obsługą magnetofonu musi zajmować się użytkownik tak samo jak mikrokomputer ZX Spectrum).

Opisana przystawka współpracuje bezawaryjnie od ponad roku z komputerem Atari 130 XE.

Spis części

- a) Rezystory typu MLT 0.125 W (lub 0.25 W)
- 1 kΩ — R 23
 - 1.5 kΩ — R 15, R 16
 - 4.7 kΩ — R 17, R 18, R 20
 - 5.6 kΩ — R 6
 - 6.8 kΩ — R 4
 - 10 kΩ — R 1, R 19, R 21, R 22
 - 5 kΩ — R 9, R 10
 - 58 kΩ — R 5
 - 68 kΩ — R 3
 - 270 kΩ — R 11, R 12, R 13, R 14
 - 1 MΩ — R 2

UWAGA:

Wartości rezystorów R 7 i R 8 ustalamy w czasie uruchamiania przystawki. W ich miejsce lutujemy potencjometry montażowe 470 k.

- b) Kondensatory typu KCP, KSF
- 1 nF — C 2, C 3, C 4, C 5, C 10, C 15
 - 6.8 nF — C 8, C 9
 - 10 nF — C 1
 - 22 nF — C 6, C 7
 - 33 nF — C 13
 - 47 nF — C 12

Kondensatory elektrolityczne

- 1 μF / 16 V — C 14
- 4.7 μF / 6.3 V — C 11

c) Półprzewodniki

- układ scalony LM 324
- BAYP17 + 19 — D 1, D 2, D 3, D 4, D 5, D 6
- BC 107 + 109 BC 237 + 239 — T 1

d) inne

wtyk magnetofonowy 5-stykowy
słyko z gniazda Eltra typu 681025 — 4 szt.
potencjometry montażowe 470 k — 2 szt.

Procedury Rom SPECTRUM (1)

Krzysztof MAMCARZ

Pojawia się coraz więcej konkursów na oprogramowanie. Wzrastają też wymagania dotyczące opracowania graficznego programu, dźwięku, szybkości działania, walorów edukacyjnych. Użytkownicy Spectrum są w gorszej sytuacji, ponieważ BASIC korzysta z niewielu możliwości systemowych i stąd tak częsta potrzeba sięgania do języka wewnętrznego. Proponujemy przedstawienie najciekawszych procedur znajdujących się w pamięci stałej ROM myśląc, że przydadzą się w programach pisanych przez Was.

Procedury te mają dwa podstawowe atuty: po pierwsze już są w pamięci, a po drugie nie zaburzają relokowalności programów w języku wewnętrznym. Każda z nich zostanie przedstawiona w postaci tabelki zawierającej numer porządkowy, nazwę (wg Complete ROM Disassembly), adres heksagonalny i dziesiętny, cel sprecyzowany w jednym zdaniu, parametry wejściowe oraz stan wyjściowy rejestrów i flag. Dokładniejsze opisy, jeśli będzie to konieczne, znajdują się pod tabelką. Pojawiające się nazwy systemowych systemowych nie będą wyjaśniane. Zainteresowanych ich znaczeniem odsyłam do wcześniejszego cyklu „Sztuki i sztuczki”. Brak uwag o stanie rejestrów oznacza, że procedura zmienia ich zawartość. Rejestr IX będzie pominięty w rozważaniach, gdyż jego zawartość powinna być stała i wynosić 23610. Jeśli procedura zmienia stan rejestrów pomocniczych, to będzie to specjalnie podkreślone. Przy powrocie do BASIC-a para HL, powinna zawsze zawierać adres 10072.

Zaczynamy od klawiatury

1	KEYBOARD	02BFh	0703
CEL:	przeszukanie klawiatury, odczyt jej stanu oraz zdekodowanie przyciśniętego klawisza według tabel translacji.		
WE:	nieistotne;		
WY:	aktualnienie zmiennych systemowych KSTATE, LAST i FLAGS (5 bit); IX przechowany.		

Procedura 1 jest osią systemu. To jedyny interfejs programowy umożliwiający komunikowanie się człowieka z komputerem. Urukochamiana jest po każdym sygnale przerwania w procedurze 38h (RST 56)

2	KEY-SCAN	028Eh	0654
CEL:	przeszukanie klawiatury		
WE:	nieistotne;		
WY:	D = 255 jeśli nie naciśnięto klawisza; 39 jeśli naciśnięto CAPS SHIFT; 24 jeśli naciśnięto SYMBOL SHIFT; E = 255 jeśli nie naciśnięto klawisza; w przeciwnym wypadku zawiera numer klawisza (0 - 39); Z — naciśnięto właściwą kombinację klawiszy; NZ — naciśnięto niewłaściwą kombinację klawiszy; IX — przechowywany.		

Przez właściwą kombinację klawiszy procedura rozumie każdy klawisz pojedynczy z pominięciem CAPS i SYMBOL SHIFT oraz każde dwa klawisze, z których jeden to CAPS lub SYMBOL SHIFT. Szczególna sytuacja zachodzi, gdy naciśnięto oba klawisze specjalne (SHIFT). Wtedy stan CAPS znajduje się w rejestrze D, a stan SYMBOL w rejestrze E.

Numerzy klawiszy są na tym poziomie rozumiane następująco

36	28	20	12	4	3	11	19	27	35
37	29	21	13	5	2	10	18	26	34
38	30	22	14	6	1	9	17	25	33
39	31	23	15	7	0	8	16	24	32

Przykład przedstawia wykorzystanie procedury do testowania wybranego klawisza.

```
10 : ** test '87
20 : GEN3M21
30      ORG      60000
40 KEYSN EQU    028Eh
50      DI
60 FRACUJ CALL  KEYSN
70      LD      A,255
80      CP      D
90      JR      NZ,FRACUJ
100     LD      A,33
110     CP      E
120     JR      NZ,FRACUJ
130     EI
140     RET
```

Pierwszą część pętli (linie 70..90) sprawdza rejestr D, a druga rejestr E (linie 100..120). Wymagane wartości wprowadzamy do akumulatora. W tym wypadku w programie testowany jest klawisz ENTER.

2	BREAK-KEY	1F54h	8020
CEL:	testowanie kombinacji CAPS SHIFT + SPACE (BREAK)		
WE:	Nieistotne;		
WY:	NC — naciśnięto BREAK; CF — nie naciśnięto BREAK;		

Testowanie BREAK może się odbywać przy wyłączonych przerwaniach. Program będzie wyglądał następująco:

```
10 : ** break '85
20 : GENA3M21
30      ORG      60000
40 BREAK EQU    1F54h
50      DI
60 TEST  CALL    BREAK
70      JR      C,TEST
```

Dekodowanie przyciśniętego klawisza jest dostosowane do warunków pracy SPECTRUM. Algorytm musi uwzględnić klawisze specjalne, odgrywające dość istotną rolę. Nic nie stoi na przeszkodzie, aby algorytm ten zastąpić i dostosować do własnych potrzeb. Co prawda, nie można zastąpić procedury 38h, ale można przełączyć tryb przerwania (IM 2).

Oto przykład innego podejścia do zagadnienia przeszukiwania i dekodowania.

```

10 : ** SCAN '87
20 : GEN$3M21
30   ORG      60000
40 U_UDG EQU  65528
50   LD       HL,U_UDG
60   LD       BC,FEFH
70 NEXT IN   A,(C)
80   CPL
90   AND      1FH
100  LD       (HL),A
110  INC      HL
120  RLC      B
130  JR       C,NEXT
140  RET

```

Odczyt stanu klawiatury następuje przez port o adresie F#H (254), który podaje się w rejestrze C. Drugi bajt, który reprezentuje jeden z ośmiu rzędów klawiatury, powiniem być podany w rejestrze B. Wtedy odczytujemy port dzięki instrukcji IN A, (C). Każdy z rzędów odpowiada więc za 5 klawiszy. Akumulator przechwytywa 5 bi-

tów, których stan jest reakcją na nasze zachowanie przy klawiaturze.

Ta metoda odczytu nie przewiduje traktowania wybranych klawiszy w jakiś specjalny sposób. Slany 40 klawiszy zostają po prostu odczytane w pamięci RAM w postaci mapy bitowej. Dalej można z nimi zrobić, co się komu podoba. I nie ma znaczenia, ile klawiszy przyciśnięto.

Dla udowodnienia tego mapa bitowa została umieszczona w obszarze grafiki użytkownika, aby można było ją oglądać. Umożliwi to następujący program:

```

10 REM ** mapa '87
20 RANDOMIZE USR 60000
30 PRINT AT 1,1:CHR$(164)
40 GOTO 20

```

W następujących odsłonach przedstawione będą procedury współpracujące z ekranem. A może znacie ciekawe rozwiązania w Spectrum, które uzupełniałyby przekazywane wiadomości? Jeśli tak, to napiszcie.

Krzysztof MAMCARZ

Tajemnice matematyki

Dzielniki liczby całkowitej

Problem znajdowania dzielników dowolnej liczby całkowitej intrzygował wielu sławnych matematyków, a ich poszukiwania wzoru opisującego liczby pierwsze długo były daremne. Kiedy wielki matematyk **Fermat**, w roku 1640 podał swoją formułę

$$2^{2^n} + 1$$

łatwo pokazał, że początkowe liczby tego rodzaju, czyli 3, 5, 17, 257, 65537 są liczbami pierwszymi. Nie potrafił jednak podać ogólnego dowodu swojej hipotezy. Trzeba było stu lat, aby inny matematyk — **Euler** udowodnił, że $2^{2^n} + 1$ nie jest liczbą pierwszą. Program nasz nie sprostaa liczbie $2^{32} + 1 = 100895598169$. Podaje jednak, dla każdej liczby mającej maksymalnie 9 cyfr, dzielniki będące liczbami pierwszymi oraz sprawdza czy dzielniki te dzielą wybraną liczbę w kolejnych swoich potęgach.

Maksymalną potęgę, przy której dzielnik podniesiony do niej dzieli podaną liczbę, nazywamy krotnością tego dzielnika. Na podstawie krotności dzielników pierwszych program wylicza liczbę wszystkich dzielników wybranej liczby, w tym liczbą 1 i podana liczba traktowane są jako dzielniki, zgodnie ze wzorem

$$(a_1 + 1) * (a_2 + 1) * \dots * (a_n + 1)$$

gdzie a_1, a_2, \dots, a_n są krotnościami dzielników pierwszych. Przykładowo $12 = 2^2 * 3$, na zatem $(2+1)*(1+1) = 6$ dzielników, są to liczby: 1, 2, 3, 4, 6, 12.

A. T.

```

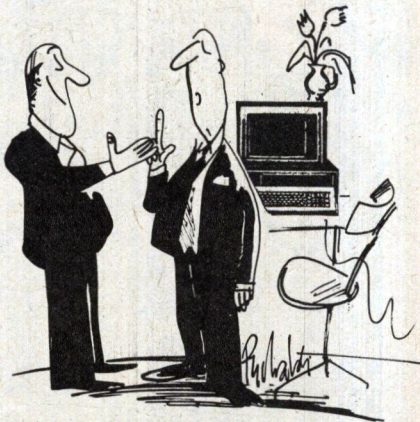
10 REM Ile dzielnikow ma liczba
15 CLS : BORDER 3
19 PRINT AT 10,8;"Liczymy (t/n
?)": BEEP .1,3
20 IF INKEY$="n" THEN STOP
25 IF INKEY$("<" THEN BEEP .2
4: GO TO 20
30 PRINT AT 10,6;"Podaj liczbę
całkowitą"
35 INPUT "a = ";a
37 CLS
40 LET a=ABS (a)
43 PRINT "Dzielnik 1"
47 LET c=a: LET (o=1
50 LET b=1+INT (a/2)
53 LET i=2 TO b
56 LET liczb=0
60 LET r=a-i*INT (a/i)

```

```

70 IF r<>0 THEN GO TO 85
75 LET liczb=liczb+1
76 LET a$="1z"
80 LET a=a/i: GO TO 85
85 IF liczb=0 THEN GO TO 90
90 PRINT "Dzielnik ";i"; k
rotnosc ";liczb: BEEP 1,3
87 LET lo=10*(liczb+1)
88 IF a=1 THEN LET i=b
89 IF i>a THEN LET i=b
90 NEXT i
95 PRINT "Dzielnik ";i
100 IF a$("<" THEN GO TO 110
105 PRINT AT 15,6;"liczba zlozo
na ";c
105 PRINT AT 15,6;"ilosc dzieln
ikow ";lo: BEEP 2,3: PRINT AT
20,3; FLASH 1;"nacisnij dowolny
klawisz": PAUSE 0: GO TO 15
110 PRINT AT 15,6;"liczba pierw
sza ";c
115 PRINT AT 15,6;"liczba dziel
nikow ";liczb: BEEP 2,3: PRINT AT 2
0,3; FLASH 1;"nacisnij dowolny
klawisz": PAUSE 0: GO TO 15

```



— A gdzie nasza, programistka?
— Komputer dał jej zwolnienie.



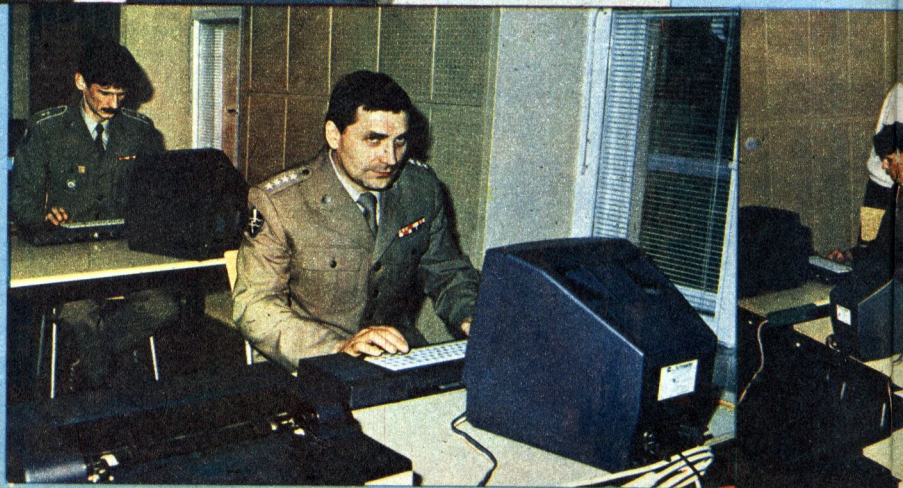
TELEB

Rozpoczęła już pracę Łączności w Zegrzu pot muszą nikogo namawiać cowni zasiedli pierwsi si mienił się w plac budow doczne.

Nowocześnie wyposaż klub mikrokomputerowy, teresuje informatyka mę

Jest w tym zasługa kr równika Ośrodka Oblicz

Wszystkim informatyko „komputerowej” działalc





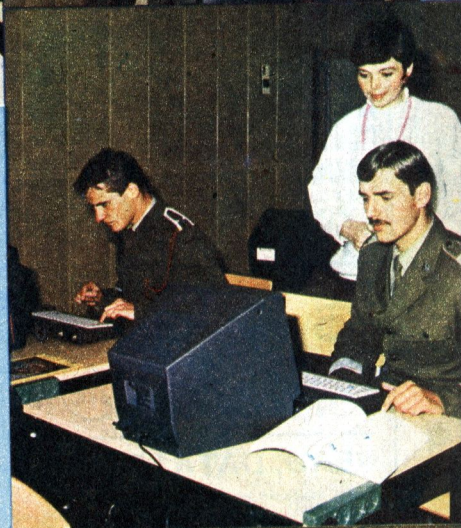
IT — ZEGRZE

Pracownia komputerowa w Wyższej Szkole Oficerskiej Wojsk
d Warszawa. Pracownicy ośrodka obliczeniowego uczelni nie
do solidnej pracy. Zresztą sami dali przykład; zanim w pra-
tuchacze przeprowadzono tu gruntowny remont. Ośrodek za-
lany, a informatycy bywało, że w budowlanców. Efekty są wi-

żona sala służyć będzie podchorążym; działać będzie również
„Telebit”. Drzwi otwarte są dla każdego — wszyscy, których in-
gą znaleźć tu swoje miejsce.

mentanta WSOWL płk. dypl. Zbigniewa Chruścińskiego i kie-
niowego mjr. mgr. inż. Tadeusza Szuszkiewicza.

om z Zegrza i sympatykom tej dyscypliny, życzymy sukcesów w
ności.



PROGRAM

28

Tkanina

Krótki program graficzny, pozwalający na otrzymanie na ekranie różnorodnych wzorów w różnych kolorach. Może to być nie tylko interesująca zabawa, ale także pomoc dla tych, którzy zajmują się projektowaniem odzieży.

Tadeusz CISEK

```

1 REM TKANINA
5 PRINTCHR*(147)
6 PRINT'TKANINA!'
-60 PRINT'WYBOR WZORU:'
70 PRINT'1) PASEK SZEROKI'
75 PRINT'2) PASEK POJEDYNCZY'
80 INPUT'3) PASEK WIELOKOLOROWY';C
90 N=INT(RND(0)*9+1)
100 FOR X=1 TO N
110 A*(X)=CHR*(INT(RND(0)*62+161))
120 NEXT X
130 PRINTCHR*(147)
140 FOR M=1TO6:IFC=1THEN GOSUB 250
150 FOR X=1TON:IFC=2 THEN GOSUB 250
160 FOR Y=1TO40:IF C=3 THEN GOSUB 250
    
```

```

170 PRINT*(X);
180 NEXT Y,X,M
210 GOTO 60
250 A=INT(RND(0)*16+1)
260 ON A GOSUB 300,310,320,330,340,350,3
50,370,380,390,400,410,420,430,440,450
    
```

```

270 RETURN
300 PRINTCHR*(5);:RETURN
310 PRINTCHR*(28);:RETURN
320 PRINTCHR*(30);:RETURN
330 PRINTCHR*(31);:RETURN
340 PRINTCHR*(129);:RETURN
350 PRINTCHR*(144);:RETURN
360 PRINTCHR*(149);:RETURN
370 PRINTCHR*(150);:RETURN
380 PRINTCHR*(151);:RETURN
390 PRINTCHR*(152);:RETURN
400 PRINTCHR*(153);:RETURN
410 PRINTCHR*(154);:RETURN
420 PRINTCHR*(155);:RETURN
430 PRINTCHR*(156);:RETURN
440 PRINTCHR*(158);:RETURN
450 PRINTCHR*(158);:RETURN
READY.
    
```

PROGRAM

29

READ, DATA i RESTORE

W kilku listach poprzedzających się prosiły o programy na mikrokomputer ZX-81, który choć zapomniany, ma widać jeszcze zwolenników. Ponieważ możliwości ZX-81 są bardzo ograniczone, może przydać się sposób symulacji komend READ, DATA i RESTORE, które implementacja BASIC'a pomija.

```

10 REM 5,-1001,0,-3,
20 LET D=0
30 LET C=0
40 LET E=16427
50 GOSUB 200
60 IF D=1 THEN LET C=-C
70 LET D=0
80 LET E=A+E+1
90 PRINT C
100 LET C=0
110 INPUT A#
120 GO TO 50
200 FOR A=0 TO 50
210 LET B=PEEK (E+A)-28
    
```

ZX 81

```

220 IF B=188 THEN RETURN
230 IF B=90 THEN STOP
240 IF NOT B=192 THEN GO TO 270
250 LET D=1
260 GO TO 280
270 LET C=C*10+B
280 NEXT A
290 STOP
    
```

Dane umieszczone są tu w postaci listy w linii 10 po instrukcji REM. Po każdej liczbie powinien nastąpić przecinek (również po ostatniej). Podprogram 200 odczytuje kolejne cyfry, a wartość liczby wkłada do zmiennej C. W ten sposób liczbę można przechwytać do własnych celów.

Symulacja komendy RESTORE polega na przypisaniu zmiennej E adresu początkowego danych, czyli adresu pierwszego znaku po REM. Dane można umieścić w dowolnej linii pod warunkiem, że znany będzie ów adres. Poniższy program wyszukuje linie z komendą REM i podaje potrzebne adresy.

```

1000 LET A=PEEK (16392)+256*PEEK (16393)
1010 FOR B=16424 TO A
1020 IF PEEK (B)=254 THEN PRINT B+1
1030 NEXT B
    
```

Jeśli REM znajduje się tylko w pierwszej linii, to wystarczy przypisać zmiennej E wartość 16427 (jak w linii 40) aby osiągnąć efekt czytania danych od początku.

K. M.

Kostka

To prosty program dla 1 lub dwóch graczy. Jeśli będzie tylko jeden, jako drugi zawodnik wystąpi komputer. Obsługa programu ogranicza się do wyboru liczby graczy, podania imion lub imienia, podania maksymalnego wyniku, do jakiego ma toczyć się gra. O tym, kto pierwszy zaczyna, decyduje przypadek i sygnalizowane to jest przez migający napis. Uzyskana ilość oczek dodawana jest do punktacji danego zawodnika. Jeśli ktoś wyrzucił „6” ma prawo powtórzyć rzut. Start „losowania” — dowolny klawisz. Komu uda się uzyskać maksymalny wynik, otrzymuje 1 duży punkt i cała zabawa zaczyna się od nowa. Przy każdej następnej partii „maksymalny wynik może (lecz nie musi) być zmieniony. Na koniec jedna uwaga, przy wpisywaniu programu należy literę „A” w liniach 1520 — 1570 wpisać w trybie „graficznym”.

Z. WIŚNIEWSKI

```

1000 REM DEFINICJA OCZEK KOSTKI
1010 DIM X(7)
1020 FOR I=0 TO 7
1030 X(I)=0
1040 NEXT I
1050 DATA 0,126,126,126,126,126,
1060 0,0,0,0,0,0,0
1070 REM IMIENIA I PLAN
1080 DIM P$(32): DIM M(1): DIM N
1090 P$="Zawodnik 1": A
1100 P$="Zawodnik 2": B
1110 INPUT "ILU graczy (1 lub 2)
1120 IF I<1 OR I>2 THEN GO TO 15
1130 IF I=1 THEN INPUT "podaj im
1140 : LET B$="Komputer"
1150 IF I=2 THEN INPUT "podaj imie
1160 : AS: INPUT "podaj imie"
1170 LET P$=P$+A
1180 LET P$=P$+B
1190 LET P$=P$+A
1200 LET P$=P$+B
1210 LET P$=P$+A
1220 LET P$=P$+B
1230 LET P$=P$+A
1240 LET P$=P$+B
1250 LET P$=P$+A
1260 LET P$=P$+B
1270 LET P$=P$+A
1280 LET P$=P$+B
1290 LET P$=P$+A
1300 LET P$=P$+B
1310 LET P$=P$+A
1320 LET P$=P$+B
1330 LET P$=P$+A
1340 LET P$=P$+B
1350 LET P$=P$+A
1360 LET P$=P$+B
1370 LET P$=P$+A
1380 LET P$=P$+B
1390 PLOT X,60
1400 PLOT X,60
1410 PLOT X,60
1420 PLOT X,60
1430 PLOT X,60
1440 PLOT X,60
1450 PLOT X,60
1460 PLOT X,60
1470 PLOT X,60
1480 PLOT X,60
1490 PLOT X,60
1500 PLOT X,60
1510 PLOT X,60
1520 GO SUB 1500
1530 PRINT AT 4,7:
1540 A=6 THEN GO TO 500
    
```

ZX SPECTRUM

```

575 IF ( )=A THEN GO TO 2500
1000 REM
1010 PRINT AT 0,0:"Zawodnik 1":
1020 BRIGHT 1: FLASH 1: AT 0,21:"Zawod
1030 nik 2"
1040 IF I=1 THEN LET D$="TERAZ I
1050 : ON I: PRINT AT 0,21:"Zawodnik 1"
1060 : ON I: PRINT AT 0,21:"Zawodnik 2"
1070 GO TO 500
1080 REM LOSOWANIE OCZEK KOSTKI
1090 LET X=1 TO 30
1100 A=INT (RND*6)+1
1110 IF A=1 THEN PRINT AT 10,14:
1120 : AT 11,14: " " : AT 12,14: "
1130 IF A=2 THEN PRINT AT 10,14:
1140 : AT 11,14: " " : AT 12,14: "
1150 IF A=3 THEN PRINT AT 10,14:
1160 : AT 11,14: " " : AT 12,14: "
1170 IF A=4 THEN PRINT AT 10,14:
1180 : AT 11,14: " " : AT 12,14: "
1190 IF A=5 THEN PRINT AT 10,14:
1200 : AT 11,14: " " : AT 12,14: "
1210 IF A=6 THEN PRINT AT 10,14:
1220 : AT 11,14: " " : AT 12,14: "
1230 BEEP 0.005*X
1240 NEXT X
1250 RETURN
1260 REM LOSOWANIE OCZEK KOSTKI
1270 LET X=1 TO 30
1280 A=INT (RND*6)+1
1290 IF A=1 THEN PRINT AT 10,14:
1300 : AT 11,14: " " : AT 12,14: "
1310 IF A=2 THEN PRINT AT 10,14:
1320 : AT 11,14: " " : AT 12,14: "
1330 IF A=3 THEN PRINT AT 10,14:
1340 : AT 11,14: " " : AT 12,14: "
1350 IF A=4 THEN PRINT AT 10,14:
1360 : AT 11,14: " " : AT 12,14: "
1370 IF A=5 THEN PRINT AT 10,14:
1380 : AT 11,14: " " : AT 12,14: "
1390 IF A=6 THEN PRINT AT 10,14:
1400 : AT 11,14: " " : AT 12,14: "
1410 BEEP 0.005*X
1420 NEXT X
1430 RETURN
1440 REM LOSOWANIE OCZEK KOSTKI
1450 LET X=1 TO 30
1460 A=INT (RND*6)+1
1470 IF A=1 THEN PRINT AT 10,14:
1480 : AT 11,14: " " : AT 12,14: "
1490 IF A=2 THEN PRINT AT 10,14:
1500 : AT 11,14: " " : AT 12,14: "
1510 IF A=3 THEN PRINT AT 10,14:
1520 : AT 11,14: " " : AT 12,14: "
1530 IF A=4 THEN PRINT AT 10,14:
1540 : AT 11,14: " " : AT 12,14: "
1550 IF A=5 THEN PRINT AT 10,14:
1560 : AT 11,14: " " : AT 12,14: "
1570 IF A=6 THEN PRINT AT 10,14:
1580 : AT 11,14: " " : AT 12,14: "
1590 BEEP 0.005*X
1600 NEXT X
1610 RETURN
1620 REM LOSOWANIE OCZEK KOSTKI
1630 LET X=1 TO 30
1640 A=INT (RND*6)+1
1650 IF A=1 THEN PRINT AT 10,14:
1660 : AT 11,14: " " : AT 12,14: "
1670 IF A=2 THEN PRINT AT 10,14:
1680 : AT 11,14: " " : AT 12,14: "
1690 IF A=3 THEN PRINT AT 10,14:
1700 : AT 11,14: " " : AT 12,14: "
1710 IF A=4 THEN PRINT AT 10,14:
1720 : AT 11,14: " " : AT 12,14: "
1730 IF A=5 THEN PRINT AT 10,14:
1740 : AT 11,14: " " : AT 12,14: "
1750 IF A=6 THEN PRINT AT 10,14:
1760 : AT 11,14: " " : AT 12,14: "
1770 BEEP 0.005*X
1780 NEXT X
1790 RETURN
1800 REM LOSOWANIE OCZEK KOSTKI
1810 LET X=1 TO 30
1820 A=INT (RND*6)+1
1830 IF A=1 THEN PRINT AT 10,14:
1840 : AT 11,14: " " : AT 12,14: "
1850 IF A=2 THEN PRINT AT 10,14:
1860 : AT 11,14: " " : AT 12,14: "
1870 IF A=3 THEN PRINT AT 10,14:
1880 : AT 11,14: " " : AT 12,14: "
1890 IF A=4 THEN PRINT AT 10,14:
1900 : AT 11,14: " " : AT 12,14: "
1910 IF A=5 THEN PRINT AT 10,14:
1920 : AT 11,14: " " : AT 12,14: "
1930 IF A=6 THEN PRINT AT 10,14:
1940 : AT 11,14: " " : AT 12,14: "
1950 BEEP 0.005*X
1960 NEXT X
1970 RETURN
1980 REM LOSOWANIE OCZEK KOSTKI
1990 LET X=1 TO 30
2000 A=INT (RND*6)+1
2010 IF A=1 THEN PRINT AT 10,14:
2020 : AT 11,14: " " : AT 12,14: "
2030 IF A=2 THEN PRINT AT 10,14:
2040 : AT 11,14: " " : AT 12,14: "
2050 IF A=3 THEN PRINT AT 10,14:
2060 : AT 11,14: " " : AT 12,14: "
2070 IF A=4 THEN PRINT AT 10,14:
2080 : AT 11,14: " " : AT 12,14: "
2090 IF A=5 THEN PRINT AT 10,14:
2100 : AT 11,14: " " : AT 12,14: "
2110 IF A=6 THEN PRINT AT 10,14:
2120 : AT 11,14: " " : AT 12,14: "
2130 BEEP 0.005*X
2140 NEXT X
2150 RETURN
2160 REM LOSOWANIE OCZEK KOSTKI
2170 LET X=1 TO 30
2180 A=INT (RND*6)+1
2190 IF A=1 THEN PRINT AT 10,14:
2200 : AT 11,14: " " : AT 12,14: "
2210 IF A=2 THEN PRINT AT 10,14:
2220 : AT 11,14: " " : AT 12,14: "
2230 IF A=3 THEN PRINT AT 10,14:
2240 : AT 11,14: " " : AT 12,14: "
2250 IF A=4 THEN PRINT AT 10,14:
2260 : AT 11,14: " " : AT 12,14: "
2270 IF A=5 THEN PRINT AT 10,14:
2280 : AT 11,14: " " : AT 12,14: "
2290 IF A=6 THEN PRINT AT 10,14:
2300 : AT 11,14: " " : AT 12,14: "
2310 BEEP 0.005*X
2320 NEXT X
2330 RETURN
2340 REM LOSOWANIE OCZEK KOSTKI
2350 LET X=1 TO 30
2360 A=INT (RND*6)+1
2370 IF A=1 THEN PRINT AT 10,14:
2380 : AT 11,14: " " : AT 12,14: "
2390 IF A=2 THEN PRINT AT 10,14:
2400 : AT 11,14: " " : AT 12,14: "
2410 IF A=3 THEN PRINT AT 10,14:
2420 : AT 11,14: " " : AT 12,14: "
2430 IF A=4 THEN PRINT AT 10,14:
2440 : AT 11,14: " " : AT 12,14: "
2450 IF A=5 THEN PRINT AT 10,14:
2460 : AT 11,14: " " : AT 12,14: "
2470 IF A=6 THEN PRINT AT 10,14:
2480 : AT 11,14: " " : AT 12,14: "
2490 BEEP 0.005*X
2500 NEXT X
2510 RETURN
2520 REM LOSOWANIE OCZEK KOSTKI
2530 LET X=1 TO 30
2540 A=INT (RND*6)+1
2550 IF A=1 THEN PRINT AT 10,14:
2560 : AT 11,14: " " : AT 12,14: "
2570 IF A=2 THEN PRINT AT 10,14:
2580 : AT 11,14: " " : AT 12,14: "
2590 IF A=3 THEN PRINT AT 10,14:
2600 : AT 11,14: " " : AT 12,14: "
2610 IF A=4 THEN PRINT AT 10,14:
2620 : AT 11,14: " " : AT 12,14: "
2630 IF A=5 THEN PRINT AT 10,14:
2640 : AT 11,14: " " : AT 12,14: "
2650 IF A=6 THEN PRINT AT 10,14:
2660 : AT 11,14: " " : AT 12,14: "
2670 BEEP 0.005*X
2680 NEXT X
2690 RETURN
2700 REM LOSOWANIE OCZEK KOSTKI
2710 LET X=1 TO 30
2720 A=INT (RND*6)+1
2730 IF A=1 THEN PRINT AT 10,14:
2740 : AT 11,14: " " : AT 12,14: "
2750 IF A=2 THEN PRINT AT 10,14:
2760 : AT 11,14: " " : AT 12,14: "
2770 IF A=3 THEN PRINT AT 10,14:
2780 : AT 11,14: " " : AT 12,14: "
2790 IF A=4 THEN PRINT AT 10,14:
2800 : AT 11,14: " " : AT 12,14: "
2810 IF A=5 THEN PRINT AT 10,14:
2820 : AT 11,14: " " : AT 12,14: "
2830 IF A=6 THEN PRINT AT 10,14:
2840 : AT 11,14: " " : AT 12,14: "
2850 BEEP 0.005*X
2860 NEXT X
2870 RETURN
2880 REM LOSOWANIE OCZEK KOSTKI
2890 LET X=1 TO 30
2900 A=INT (RND*6)+1
2910 IF A=1 THEN PRINT AT 10,14:
2920 : AT 11,14: " " : AT 12,14: "
2930 IF A=2 THEN PRINT AT 10,14:
2940 : AT 11,14: " " : AT 12,14: "
2950 IF A=3 THEN PRINT AT 10,14:
2960 : AT 11,14: " " : AT 12,14: "
2970 IF A=4 THEN PRINT AT 10,14:
2980 : AT 11,14: " " : AT 12,14: "
2990 IF A=5 THEN PRINT AT 10,14:
3000 : AT 11,14: " " : AT 12,14: "
3010 IF A=6 THEN PRINT AT 10,14:
3020 : AT 11,14: " " : AT 12,14: "
3030 BEEP 0.005*X
3040 NEXT X
3050 RETURN
3060 REM LOSOWANIE OCZEK KOSTKI
3070 LET X=1 TO 30
3080 A=INT (RND*6)+1
3090 IF A=1 THEN PRINT AT 10,14:
3100 : AT 11,14: " " : AT 12,14: "
3110 IF A=2 THEN PRINT AT 10,14:
3120 : AT 11,14: " " : AT 12,14: "
3130 IF A=3 THEN PRINT AT 10,14:
3140 : AT 11,14: " " : AT 12,14: "
3150 IF A=4 THEN PRINT AT 10,14:
3160 : AT 11,14: " " : AT 12,14: "
3170 IF A=5 THEN PRINT AT 10,14:
3180 : AT 11,14: " " : AT 12,14: "
3190 IF A=6 THEN PRINT AT 10,14:
3200 : AT 11,14: " " : AT 12,14: "
3210 BEEP 0.005*X
3220 NEXT X
3230 RETURN
3240 REM LOSOWANIE OCZEK KOSTKI
3250 LET X=1 TO 30
3260 A=INT (RND*6)+1
3270 IF A=1 THEN PRINT AT 10,14:
3280 : AT 11,14: " " : AT 12,14: "
3290 IF A=2 THEN PRINT AT 10,14:
3300 : AT 11,14: " " : AT 12,14: "
3310 IF A=3 THEN PRINT AT 10,14:
3320 : AT 11,14: " " : AT 12,14: "
3330 IF A=4 THEN PRINT AT 10,14:
3340 : AT 11,14: " " : AT 12,14: "
3350 IF A=5 THEN PRINT AT 10,14:
3360 : AT 11,14: " " : AT 12,14: "
3370 IF A=6 THEN PRINT AT 10,14:
3380 : AT 11,14: " " : AT 12,14: "
3390 BEEP 0.005*X
3400 NEXT X
3410 RETURN
3420 REM LOSOWANIE OCZEK KOSTKI
3430 LET X=1 TO 30
3440 A=INT (RND*6)+1
3450 IF A=1 THEN PRINT AT 10,14:
3460 : AT 11,14: " " : AT 12,14: "
3470 IF A=2 THEN PRINT AT 10,14:
3480 : AT 11,14: " " : AT 12,14: "
3490 IF A=3 THEN PRINT AT 10,14:
3500 : AT 11,14: " " : AT 12,14: "
3510 IF A=4 THEN PRINT AT 10,14:
3520 : AT 11,14: " " : AT 12,14: "
3530 IF A=5 THEN PRINT AT 10,14:
3540 : AT 11,14: " " : AT 12,14: "
3550 IF A=6 THEN PRINT AT 10,14:
3560 : AT 11,14: " " : AT 12,14: "
3570 BEEP 0.005*X
3580 NEXT X
3590 RETURN
3600 REM LOSOWANIE OCZEK KOSTKI
3610 LET X=1 TO 30
3620 A=INT (RND*6)+1
3630 IF A=1 THEN PRINT AT 10,14:
3640 : AT 11,14: " " : AT 12,14: "
3650 IF A=2 THEN PRINT AT 10,14:
3660 : AT 11,14: " " : AT 12,14: "
3670 IF A=3 THEN PRINT AT 10,14:
3680 : AT 11,14: " " : AT 12,14: "
3690 IF A=4 THEN PRINT AT 10,14:
3700 : AT 11,14: " " : AT 12,14: "
3710 IF A=5 THEN PRINT AT 10,14:
3720 : AT 11,14: " " : AT 12,14: "
3730 IF A=6 THEN PRINT AT 10,14:
3740 : AT 11,14: " " : AT 12,14: "
3750 BEEP 0.005*X
3760 NEXT X
3770 RETURN
3780 REM LOSOWANIE OCZEK KOSTKI
3790 LET X=1 TO 30
3800 A=INT (RND*6)+1
3810 IF A=1 THEN PRINT AT 10,14:
3820 : AT 11,14: " " : AT 12,14: "
3830 IF A=2 THEN PRINT AT 10,14:
3840 : AT 11,14: " " : AT 12,14: "
3850 IF A=3 THEN PRINT AT 10,14:
3860 : AT 11,14: " " : AT 12,14: "
3870 IF A=4 THEN PRINT AT 10,14:
3880 : AT 11,14: " " : AT 12,14: "
3890 IF A=5 THEN PRINT AT 10,14:
3900 : AT 11,14: " " : AT 12,14: "
3910 IF A=6 THEN PRINT AT 10,14:
3920 : AT 11,14: " " : AT 12,14: "
3930 BEEP 0.005*X
3940 NEXT X
3950 RETURN
3960 REM LOSOWANIE OCZEK KOSTKI
3970 LET X=1 TO 30
3980 A=INT (RND*6)+1
3990 IF A=1 THEN PRINT AT 10,14:
4000 : AT 11,14: " " : AT 12,14: "
4010 IF A=2 THEN PRINT AT 10,14:
4020 : AT 11,14: " " : AT 12,14: "
4030 IF A=3 THEN PRINT AT 10,14:
4040 : AT 11,14: " " : AT 12,14: "
4050 IF A=4 THEN PRINT AT 10,14:
4060 : AT 11,14: " " : AT 12,14: "
4070 IF A=5 THEN PRINT AT 10,14:
4080 : AT 11,14: " " : AT 12,14: "
4090 IF A=6 THEN PRINT AT 10,14:
4100 : AT 11,14: " " : AT 12,14: "
4110 BEEP 0.005*X
4120 NEXT X
4130 RETURN
4140 REM LOSOWANIE OCZEK KOSTKI
4150 LET X=1 TO 30
4160 A=INT (RND*6)+1
4170 IF A=1 THEN PRINT AT 10,14:
4180 : AT 11,14: " " : AT 12,14: "
4190 IF A=2 THEN PRINT AT 10,14:
4200 : AT 11,14: " " : AT 12,14: "
4210 IF A=3 THEN PRINT AT 10,14:
4220 : AT 11,14: " " : AT 12,14: "
4230 IF A=4 THEN PRINT AT 10,14:
4240 : AT 11,14: " " : AT 12,14: "
4250 IF A=5 THEN PRINT AT 10,14:
4260 : AT 11,14: " " : AT 12,14: "
4270 IF A=6 THEN PRINT AT 10,14:
4280 : AT 11,14: " " : AT 12,14: "
4290 BEEP 0.005*X
4300 NEXT X
4310 RETURN
4320 REM LOSOWANIE OCZEK KOSTKI
4330 LET X=1 TO 30
4340 A=INT (RND*6)+1
4350 IF A=1 THEN PRINT AT 10,14:
4360 : AT 11,14: " " : AT 12,14: "
4370 IF A=2 THEN PRINT AT 10,14:
4380 : AT 11,14: " " : AT 12,14: "
4390 IF A=3 THEN PRINT AT 10,14:
4400 : AT 11,14: " " : AT 12,14: "
4410 IF A=4 THEN PRINT AT 10,14:
4420 : AT 11,14: " " : AT 12,14: "
4430 IF A=5 THEN PRINT AT 10,14:
4440 : AT 11,14: " " : AT 12,14: "
4450 IF A=6 THEN PRINT AT 10,14:
4460 : AT 11,14: " " : AT 12,14: "
4470 BEEP 0.005*X
4480 NEXT X
4490 RETURN
4500 REM LOSOWANIE OCZEK KOSTKI
4510 LET X=1 TO 30
4520 A=INT (RND*6)+1
4530 IF A=1 THEN PRINT AT 10,14:
4540 : AT 11,14: " " : AT 12,14: "
4550 IF A=2 THEN PRINT AT 10,14:
4560 : AT 11,14: " " : AT 12,14: "
4570 IF A=3 THEN PRINT AT 10,14:
4580 : AT 11,14: " " : AT 12,14: "
4590 IF A=4 THEN PRINT AT 10,14:
4600 : AT 11,14: " " : AT 12,14: "
4610 IF A=5 THEN PRINT AT 10,14:
4620 : AT 11,14: " " : AT 12,14: "
4630 IF A=6 THEN PRINT AT 10,14:
4640 : AT 11,14: " " : AT 12,14: "
4650 BEEP 0.005*X
4660 NEXT X
4670 RETURN
4680 REM LOSOWANIE OCZEK KOSTKI
4690 LET X=1 TO 30
4700 A=INT (RND*6)+1
4710 IF A=1 THEN PRINT AT 10,14:
4720 : AT 11,14: " " : AT 12,14: "
4730 IF A=2 THEN PRINT AT 10,14:
4740 : AT 11,14: " " : AT 12,14: "
4750 IF A=3 THEN PRINT AT 10,14:
4760 : AT 11,14: " " : AT 12,14: "
4770 IF A=4 THEN PRINT AT 10,14:
4780 : AT 11,14: " " : AT 12,14: "
4790 IF A=5 THEN PRINT AT 10,14:
4800 : AT 11,14: " " : AT 12,14: "
4810 IF A=6 THEN PRINT AT 10,14:
4820 : AT 11,14: " " : AT 12,14: "
4830 BEEP 0.005*X
4840 NEXT X
4850 RETURN
4860 REM LOSOWANIE OCZEK KOSTKI
4870 LET X=1 TO 30
4880 A=INT (RND*6)+1
4890 IF A=1 THEN PRINT AT 10,14:
4900 : AT 11,14: " " : AT 12,14: "
4910 IF A=2 THEN PRINT AT 10,14:
4920 : AT 11,14: " " : AT 12,14: "
4930 IF A=3 THEN PRINT AT 10,14:
4940 : AT 11,14: " " : AT 12,14: "
4950 IF A=4 THEN PRINT AT 10,14:
4960 : AT 11,14: " " : AT 12,14: "
4970 IF A=5 THEN PRINT AT 10,14:
4980 : AT 11,14: " " : AT 12,14: "
4990 IF A=6 THEN PRINT AT 10,14:
5000 : AT 11,14: " " : AT 12,14: "
5010 BEEP 0.005*X
5020 NEXT X
5030 RETURN
5040 REM LOSOWANIE OCZEK KOSTKI
5050 LET X=1 TO 30
5060 A=INT (RND*6)+1
5070 IF A=1 THEN PRINT AT 10,14:
5080 : AT 11,14: " " : AT 12,14: "
5090 IF A=2 THEN PRINT AT 10,14:
5100 : AT 11,14: " " : AT 12,14: "
5110 IF A=3 THEN PRINT AT 10,14:
5120 : AT 11,14: " " : AT 12,14: "
5130 IF A=4 THEN PRINT AT 10,14:
5140 : AT 11,14: " " : AT 12,14: "
5150 IF A=5 THEN PRINT AT 10,14:
5160 : AT 11,14: " " : AT 12,14: "
5170 IF A=6 THEN PRINT AT 10,14:
5180 : AT 11,14: " " : AT 12,14: "
5190 BEEP 0.005*X
5200 NEXT X
5210 RETURN
5220 REM LOSOWANIE OCZEK KOSTKI
5230 LET X=1 TO 30
5240 A=INT (RND*6)+1
5250 IF A=1 THEN PRINT AT 10,14:
5260 : AT 11,14: " " : AT 12,14: "
5270 IF A=2 THEN PRINT AT 10,14:
5280 : AT 11,14: " " : AT 12,14: "
5290 IF A=3 THEN PRINT AT 10,14:
5300 : AT 11,14: " " : AT 12,14: "
5310 IF A=4 THEN PRINT AT 10,14:
5320 : AT 11,14: " " : AT 12,14: "
5330 IF A=5 THEN PRINT AT 10,14:
5340 : AT 11,14: " " : AT 12,14: "
5350 IF A=6 THEN PRINT AT 10,14:
5360 : AT 11,14: " " : AT 12,14: "
5370 BEEP 0.005*X
5380 NEXT X
5390 RETURN
5400 REM LOSOWANIE OCZEK KOSTKI
5410 LET X=1 TO 30
5420 A=INT (RND*6)+1
5430 IF A=1 THEN PRINT AT 10,14:
5440 : AT 11,14: " " : AT 12,14: "
5450 IF A=2 THEN PRINT AT 10,14:
5460 : AT 11,14: " " : AT 12,14: "
5470 IF A=3 THEN PRINT AT 10,14:
5480 : AT 11,14: " " : AT 12,14: "
5490 IF A=4 THEN PRINT AT 10,14:
5500 : AT 11,14: " " : AT 12,14: "
5510 IF A=5 THEN PRINT AT 10,14:
5520 : AT 11,14: " " : AT 12,14: "
5530 IF A=6 THEN PRINT AT 10,14:
5540 : AT 11,14: " " : AT 12,14: "
5550 BEEP 0.005*X
5560 NEXT X
5570 RETURN
5580 REM LOSOWANIE OCZEK KOSTKI
5590 LET X=1 TO 30
5600 A=INT (RND*6)+1
5610 IF A=1 THEN PRINT AT 10,14:
5620 : AT 11,14: " " : AT 12,14: "
5630 IF A=2 THEN PRINT AT 10,14:
5640 : AT 11,14: " " : AT 12,14: "
5650 IF A=3 THEN PRINT AT 10,14:
5660 : AT 11,14: " " : AT 12,14: "
5670 IF A=4 THEN PRINT AT 10,14:
5680 : AT 11,14: " " : AT 12,14: "
5690 IF A=5 THEN PRINT AT 10,14:
5700 : AT 11,14: " " : AT 12,14: "
5710 IF A=6 THEN PRINT AT 10,14:
5720 : AT 11,14: " " : AT 12,14: "
5730 BEEP 0.005*X
5740 NEXT X
5750 RETURN
5760 REM LOSOWANIE OCZEK KOSTKI
5770 LET X=1 TO 30
5780 A=INT (RND*6)+1
5790 IF A=1 THEN PRINT AT 10,14:
5800 : AT 11,14: " " : AT 12,14: "
5810 IF A=2 THEN PRINT AT 10,14:
5820 : AT 11,14: " " : AT 12,14: "
5830 IF A=3 THEN PRINT AT 10,14:
5840 : AT 11,14: " " : AT 12,14: "
5850 IF A=4 THEN PRINT AT 10,14:
5860 : AT 11,14: " " : AT 12,14: "
5870 IF A=5 THEN PRINT AT 10,14:
5880 : AT 11,14: " " : AT 12,14: "
5890 IF A=6 THEN PRINT AT 10,14:
5900 : AT 11,14: " " : AT 12,14: "
5910 BEEP 0.005*X
5920 NEXT X
5930 RETURN
5940 REM LOSOWANIE OCZEK KOSTKI
5950 LET X=1 TO 30
5960 A=INT (RND*6)+1
5970 IF A=1 THEN PRINT AT 10,14:
5980 : AT 11,14: " " : AT 12,14: "
5990 IF A=2 THEN PRINT AT 10,14:
6000 : AT 11,14: " " : AT 12,14: "
6010 IF A=3 THEN PRINT AT 10,14:
6020 : AT 11,14: " " : AT 12,14: "
6030 IF A=4 THEN PRINT AT 10,14:
6040 : AT 11,14: " " : AT 12,14: "
6050 IF A=5 THEN PRINT AT 10,14:
6060 : AT 11,14: " " : AT 12,14: "
6070 IF A=6 THEN PRINT AT 10,14:
6080 : AT 11,14: " " : AT 12,14: "
6090 BEEP 0.005*X
6100 NEXT X
6110 RETURN
6120 REM LOSOWANIE OCZEK KOSTKI
6130 LET X=1 TO 30
6140 A=INT (RND*6)+1
6150 IF A=1 THEN PRINT AT 10,14:
6160 : AT 11,14: " " : AT 12,14: "
6170 IF A=2 THEN PRINT AT 10,14:
6180 : AT 11,14: " " : AT 12,14: "
6190 IF A=3 THEN PRINT AT 10,14:
6200 : AT 11,14: " " : AT 12,14: "
6210 IF A=4 THEN PRINT AT 10,14:
6220 : AT 11,14: " " : AT 12,14: "
6230 IF A=5 THEN PRINT AT 10,14:
6240 : AT 11,14: " " : AT 12,14: "
6250 IF A=6 THEN PRINT AT 10,14:
6260 : AT 11,14: " " : AT 12,14: "
6270 BEEP 0.005*X
6280 NEXT X
6290 RETURN
6300 REM LOSOWANIE OCZEK KOSTKI
6310 LET X=1 TO 30
6320 A=INT (RND*6)+1
6330 IF A=1 THEN PRINT AT 10,14:
6340 : AT 11,14: " " : AT 12,14: "
6350 IF A=2 THEN PRINT AT 10,14:
6360 : AT 11,14: " " : AT 12,14: "
6370 IF A=3 THEN PRINT AT 10,14:
6380 : AT 11,14: " " : AT 12,14: "
6390 IF A=4 THEN PRINT AT 10,14:
6400 : AT 11,14: " " : AT 12,14: "
6410 IF A=5 THEN PRINT AT 10,14:
6420 : AT 11,14: " " : AT 12,14: "
6430 IF A=6 THEN PRINT AT 10,14:
6440 : AT 11,14: " " : AT 12,14: "
6450 BEEP 0.005*X
6460 NEXT X
6470 RETURN
6480 REM LOSOWANIE OCZEK KOSTKI
6490 LET X=1 TO 30
6500 A=INT (RND*6)+1
6510 IF A=1 THEN PRINT AT 10,14:
6520 : AT 11,14: " " : AT 12,14: "
6530 IF A=2 THEN PRINT AT 10,14:
6540 : AT 11,14: " " : AT 12,14: "
6550 IF A=3 THEN PRINT AT 10,14:
6560 : AT 11,14: " " : AT 12,14: "
6570 IF A=4 THEN PRINT AT 10,14:
6580 : AT 11,14: " " : AT 12,14: "
6590 IF A=5 THEN PRINT AT 10,14:
6600 : AT 11,14: " " : AT 12,14: "
6610 IF A=6 THEN PRINT AT 10,14:
6620 : AT 11,14: " " : AT 12,14: "
6630 BEEP 0.005*X
6640 NEXT X
6650 RETURN
6660 REM LOSOWANIE OCZEK KOSTKI
6670 LET X=1 TO 30
6680 A=INT (RND*6)+1
6690 IF A=1 THEN PRINT AT 10,14:
6700 : AT 11,14: " " : AT 12,14: "
6710 IF A=2 THEN PRINT AT 10,14:
6720 : AT 11,14: " " : AT 12,14: "
6730 IF A=3 THEN PRINT AT 10,14:
6740 : AT 11,14: " " : AT 12,14: "
6750 IF A=4 THEN PRINT AT 10,14:
6760 : AT 11,14: " " : AT 12,14: "
6770 IF A=5 THEN PRINT AT 10,14:
6780 : AT 11,14: " " : AT 12,14: "
6790 IF A=6 THEN PRINT AT 10,14:
6800 : AT 11,14: " " : AT 12,14: "
6810 BEEP 0.005*X
6820 NEXT X
6830 RETURN
6840 REM LOSOWANIE OCZEK KOSTKI
6850 LET X=1 TO 30
6860 A=INT (RND*6)+1
6870 IF A=
```

Opowiadaniem tym rozpoczynamy prezentację utworów, nagrodzonych i wyróżnionych w naszym konkursie „Przed nami XXI wiek”. „Final”, jak Czytelnicy zapewne pamiętają, uzyskał równorzędną, trzecią nagrodę.

Korzystając ze sposobności, uprzejmie prosimy Autorów nagrodzonych i wyróżnionych prac o wyrozumiałość — i zgodę na redakcyjne opracowanie oraz skróty, dokonywane w Ich tekstach. Przyczyna — prosta: brak miejsca. W „IKS-ie” możemy publikować teksty o maksymalnej objętości do 6 stron normalizowanego maszynopisu. Ze swej strony — dołożymy wszelkich starań, aby nic nie uronić z artystycznej wartości Waszych prac, Szanowni Autorzy.

FINAL

Sorg stał zadumany na płaskim brzegu metanowego morza, planety Yarr, jak ją sobie ochrzcił. Stąd mógł najlepiej obserwować zachód ogromnego czerwonego olbrzyma, gwiazdy tej planety. Był to widok pełen majestatu i jakiejś grozy. Być może nastroje to podtrzymywała świadomość, że w tym kosmosie jest to ostatnia istniejąca gwiazda, ostatnia planeta. A sam Sorg?

Ostatnia iskierka świadomej materii. Wielkie wyzwanie rzucone śmierci przez dawno wymarłą cywilizację.

Dawno wymarła... jak dawno? Dla Sorga nie było to specjalnie istotne. Już teraz nie. W każdym razie były to miliardy lat. Lat ziemskich, liczonych czasem obiegu Ziemi, kolebki cywilizacji, wokół Słońca, niepozornej gwiazdy tej planety. Trzy, cztery... sześć miliardów lat? Rzecz bez znaczenia.

Od chwili śmierci ostatnich ludzi czas Sorga liczył się miarą rozszerzającego się stale kosmosu, rozrzedzającego się materii. Zwiększały się odległości między galaktykami, między gwiazdami, pękła iluzoryczna homeostaza systemów planetarnych. Energia rozpraszala się w narastającym tempie. Dokonały żywota galaktyki i gwiazdy, a ich materia rozprzyszyła się w przestrzeni, stale rosnącej przestrzeni...

Teraz był już tylko Sorg i to ostatnie wielkie czerwone słońce, które także niebawem dokona żywota, wybuchając supernową, rozproszy większość swej materii, jeżeli niecała i w ogniu tego ostatniego spektaklu przyrody spali się też Yarr...

Sorg przeklinałby swych twórców, gdyby mógł czuć cokolwiek. Jego świadomość była czysta, bez biochemicznych wyznaczników osobowości. Nie miał temperamentu, nie odczuwał gniewu, nienawiści czy miłości... Lecz przecież ciągle towarzyszyła mu jakaś nieuchwytna forma cierpienia.

Wiedział, że to niemożliwe, a jednak... chwycił się też często na tym, że z pewną, hmm, tęsknotą (o ile to możliwe) myślał o śmierci, zgaśnięciu... absolutnym wyłączeniu się. To jednak było niemożliwe. Na razie. Nie spełnił jeszcze swego posłannictwa, którego sam się przecież zupełnie dobrowolnie podjął... Czy je spełni? I na co się to zda?

Słońce zaszło. Niebo na zachodzie grało jeszcze blaskami czerwieni. Sorg ruszył w zamysleniu wzdłuż brzegu. Jego osiem pąjących nóg pracowało cierpliwie bez udziału woli. Czuł wielkie znużenie, choć i to było przecież niemożliwe. Był najdoskonalszym chyba wytworem swej cywilizacji; wszystkie ogromne możliwości w okresie jej największego rozkwitu sprzegły się, by mógł powstać. Od tego czasu istnieje: zwyczajny człowiek, jakim był Sorg Altenberg — stał się Tym, Który Przetrwał. Był praktycznie wieczny, niezależny od czasu, przestrzeni, energii.

Pamiętał erę wielkości cywilizacji, jedynej (jak się okazało) rozumnej formy życia w kosmosie. Wspaniałe osiągnięcia myśli, cuda techniki; on sam zresztą był takim największym cudem. Był to czas rozkwitu, ekspansji, adaptacji coraz to nowych światów, coraz to nowych galaktyk do potrzeb człowieka.

Potem nieuchronnie nadszedł czas regresji starości i śmierci gatunku homo. Prawa Biologii okazały się nieublagane. „Cokolwiek rozkwita, niezawodnie szczytnie...” — pamiętał słowa jakiegoś barbarzyńskiego poety. Potem mógł już tylko oglądać, jak rozpuszypie się w pył to, co cywilizacja stworzyła.

Był jeszcze on sam. Rozpaczliwy protest Rozumu przeciwko przemianom. Tylko po co, jaki może być

sens w tym przedstawieniu, które za-inscenizowała Przyroda? Krzyżowana przestrzeń wszechświata okazała się ujemna. Oznaczało to ciągle, nieskończone rozszerzanie się, rozpraszanie się energii. Nie było sensu czekać tu dłużej. Sorg zdecydował się na opuszczenie Yarr. Czas był już najwyższy, niebawem nawet dla niego stałoby się tu nieprzyjemnie. Skoczył na bezpieczną odległość kilkunastu lat świetlnych w pustą przestrzeń i stąd mógł obserwować ostatni akt tej sztuki.

Czerwony olbrzym gwałtownie skurczył się, zmieniając w oślepiająco jasną kulę, która następnie gwałtownie zaczęła się rozszerzać. Sorg widział nieskończenie wiele takich spektakli, zawsze jednak chętnie je oglądał. A ten, ostatni, miał swój specyficzny nastrój. Rozprzyszały się resztki skoncentrowanej energii, resztki światła...

Teraz i zawsze już tylko pustka. Pustka i on, Sorg, już teraz w ostateczny sposób absolutnie sam. Na zawsze... Wtedy pojął, zrozumiał. Jak westchnienie ulgi spłynęło na niego zrozumienie. W tej chwili wypełnił swoje posłannictwo. Zrozumiał, że to, co usiłowali pojąć najwięksi medycy, z czym biedziły się niezliczone pokolenia uczonych, filozofów, mistyków wreszcie, poszukując sensu, najogólniejszych prawideł, budując modele, systemy praw, teorii mniej lub bardziej spójnych... Zrozumiał, że tu sensu po prostu nie ma i nie było. Wszystko było bezsensowne!

Sorg bez wahania uruchomił mechanizm samozniszczenia. Dookołań ciemność rozdarł na krótką chwilę jeden jasny błysk.

Andrzej DYRSKA

ATARI STAR TEXTER *****

W IK5-ie nr 5 z bieżącego roku zaprezentowany został zestaw z wielu opracowanych dla mikrokomputerów ATARI edytorów tekstu - ATARI WRITER i w szponach ATARI, Atari 800. Autorzy artykułu stwierdzili, że powinna wada tego programu jest brak możliwości uzyskiwania znaków specyficznych dla polskiego alfabetu zarówno na ekranie jak i na dyskietce. Rzeczywiście dla polskiego użytkownika jest to bardzo istotna wada, eliminująca praktycznie ten program jako narzędzie do redagowania i drukowania artykułów, pism, itp.

Chciałbym przedstawić dzisiaj czytelnikom inny edytor tekstu STAR TEXTER, wprowadzony na rynek oprogramowania w 1985 roku przez firmę SYBEX Verlag GmbH. Autorem programu jest Toni Schwaiger, a wersję programu na ATARI opracował Carsten Reinhardt. Najważniejszą zaletą tego edytora, w stosunku do takich jak ATARI WRITER, jest to, że umożliwia on w łatwy sposób definiować i wykorzystywać, praktycznie dowolny kształt liter, które są wyświetlane na monitorze i drukowane na drukarce, przy czym wszystko to dzieje się bez jakiegokolwiek ingerencji w konstrukcję programu jak i współpracującej z nim drukarki.

Na początku kilka informacji ogólnych o typie edytora STAR TEXTER do prawidłowego funkcjonowania wymaga stacji dysków. Bufor pamięci, w którym przechowywany jest redagowany tekst, posiada pojemność ok. 20 kB. Jest on zorganizowany w 256 linii po 80 znaków każda, co umożliwia obróbkę tekstu o objętości 7-8 stron maszynopisu formatu A-4. Teksty o większej objętości muszą być dzielone na części, zapisywane na dyskietce, które są redagowane oraz drukowane oddzielnie. Edytor STAR TEXTER składa się z trzech zasadniczych programów użytkowych: programu instalacyjnego o nazwie INSTALL.BAS, właściwego edytora oraz programu umożliwiającego definiowanie potrzebnych użytkownikowi znaków. Program ten napisany jest w języku BASIC i zapisany jest w zbiorze o nazwie STARFONT.BAS. Ponieważ dzięki temu programowi STAR TEXTER posiada cechy tak istotnie, z punktu widzenia polskiego użytkownika, wyróżniająca go spośród innych

edytorów, warto chyba przedstawić całego edytora rozpoczynając od niego.

STARFONT.BAS jest programem, dzięki któremu użytkownik może definiować własne zestawy znaków, bądź modyfikować już istniejące. Możliwość ta może być specyficzna znaki dla polskiego alfabetu lub inne. Na przykład: matematyk bez problemu może pisać takie wzory:

$$\begin{matrix} 100 & b \\ E(n) & \text{lub } f(x) \\ n=1 & a \end{matrix}$$

Ktoś, kto bardzo lubi skróty, może w redagowanym piśmie powołać swego numer telefonu poprzedzony znakiem # zamiast skrótem "tel.". Natomiast ten, kto ma przyjaciela w Moskwie, może za pomocą tego edytora na ekranie w liście zaczynający się tak: Дороча Бопс. ... Praktycznie kształt definiowanych znaków zależy jedynie od wyobraźni użytkownika i oczywiście od możliwości komputera. Tworzony znak musi się mieścić w matrycy 8x8 pikseli - co może niezadowolnie zwolenników kaligrafii, lecz proszę pamiętać, że ATARI jest tylko mikrokomputerem domowym. Na dyskietce zawierającej STAR TEXTER można zdefiniować 10 różnych zestawów znaków. Jeżeli liczba ta nie zaszkodzi potrzebę użytkownika, można na kolejnej dyskietce zdefiniować i zapisać kolejne jeszcze dziesięć zestawów.

Drugą ceną zalety edytora STAR TEXTER jest możliwość wydrukowania go do współpracy z dowolnym typem drukarki, która może podłączyć do mikrokomputera ATARI. Standardowo, edytor zawarty na dyskietce umożliwia wybranie jednej z czterech drukarek:

1. ATARI 1020
2. ATARI 1025

3. EPSON (lub inna) z interfejsem z EPSON (lub inna) z łączącymi wykorzystującym głazdki do złączek. Zadanie przystosowania edytora do współpracy z drukarkami innymi niż wyżej wymienione realizuje program INSTALL.BAS. Po jego uruchomieniu wyświetlane są cztery typy drukarek jak wyżej. Jeżeli użytkownik nie posiada jednej z czterech wymienionych drukarek, w zależności od typu drukarki, którym drukarka połączona jest z komputerem (interfejs lub Game Port), musi on wybrać liczbę 3 lub 4. Następnie należy podać adres drukarki. Będąc seksem, podajemy liczbę dziesiętną wywołującą określoną funkcję drukarki. Na przykład: jeżeli drukarka

użytkownika łączy podkreślenie, otrzymując sekwencje znaków ESC,CONTROL,F, to w programie INSTALL.BAS odpowiednia funkcja włączająca ją podkreślenie musi być wywołana sekwencją kodów 27,25. Program INSTALL.BAS umożliwia zdefiniowanie, poprzez podanie sekwencji kodów, maksymalnie 16 funkcji ponumerowanych od 0 do 9. W redagowanym tekście można używać funkcji - przez podanie odpowiadających im liczb w celu wyróżnienia jego fragmentów. Wtedy edytor stwarza użytkownikowi możliwości redagowania tekstu w dowolnych, które oferuje ATARI WRITER. Można w redagowanym tekście wstawiać lub kasować znaki, wyrazy, zdania i całe paragrafy. Można kopiować całe fragmenty tekstu, wstawiać je w inne, wyrównywać marginesy lub nie, można zlecić edytorowi wyszukanie w napisanym tekście określonych wyrazów i zastąpić je innymi. W ten sposób i w ten sposób, w ten sposób współpracującej z komputerem można fragmenty tekstu wyróżniać przez drukowanie ich innym kolorem pisma. Na przykład: pracując z drukarką ATARI 1020 można stosować rozszerzone. Przy wykorzystaniu innych drukarek można pisać piśmem wysokim, niskim, podniesionym lub opuszczonym. Po napisaniu tekstu można zlecić edytorowi jego sformatowanie, wówczas zostanie on przedstawiony na ekranie dokładnie w takiej postaci w jakiej będzie później zostanie wydrukowany. Do obejrzenia tekstu sformatowanego służy zawarta w edytorze opcja przedstawiająca tekst w trybie 80 kolumnowym - oznacza to, że w edytorze nie ma wyświetlanych jest 60 znaków. Tryb ten jest bardzo wygodny dla użytkownika, ponieważ widnie potwierdza kształt tekstu, jaki wówczas ma być wydrukowany, eliminując jednocześnie nieudne przesuwanie kursora od lewego marginesu do prawego i odwrotnie. Tekst przedstawiony w trybie 80 kolumnowym jest trudniejszy do czytania, z tego względu jest normalnie znak (wyświetlany w trybie 40 kolumnowym) składa się z 64 pikseli, natomiast w tym trybie jest on "przepołowiany" - składa się z 32 pikseli (4x8).

Inną ceną zalety edytora jest to, że oferuje on użytkownikowi w trakcie pisania tekstu wykorzystanie, czasami bardzo potrzebnego narzędzia - kalkulatora. Jeżeli w trakcie pisania użytkownik potrzebuje obliczyć wartość skomplikowanego wyrażenia matematycznego, nie musi sięgać po kartkę, ołówek czy elektroniczny kalkulator. Wystarczy w jakikolwiek momencie wpisać to wyrażenie tak jak się pisze wyrażenie matematyczne w języku BASIC, wywoła kalkulator edytora i za chwilkę otrzymamy wyświetlony na ekranie wynik. Kalkulator w STAR TEXTERze dopuszcza wszystkie pięć operacji, to jest: odejmowania, dodawania, mnożenia, dzielenia i potęgowania. Oprócz tego TEXTER jest interesujący dla polskiego użytkownika ze względu na dostępność "polskich" znaków zarówno na ekranie jak i drukarce. Brak możliwości STAR TEXTERA obrazuje, że nie jest to narzędzie, którego redagowania i wydrukowania na drukarce ATARI 1020 przy jego pomocy.

Przedstawiając czytelnikom IK5-a jeszcze jeden edytor tekstów opracowany dla mikrokomputerów ATARI, nie chcę stwierdzić, że jest to narzędzie lepsze od innych. Naprawdę STAR TEXTER jest interesujący dla polskiego użytkownika ze względu na dostępność "polskich" znaków zarówno na ekranie jak i drukarce. Brak możliwości STAR TEXTERA obrazuje, że nie jest to narzędzie, którego redagowania i wydrukowania na drukarce ATARI 1020 przy jego pomocy.

Redakcja dziękuje Towarzystwu Wynalazców Ratio za udostępnienie polskiej wersji edytora Star-Texter, którego możliwości zaprezentowaliśmy powyżej.

BASIC (9)

W tym wykładzie podamy podstawowe wiadomości o tablicach zmiennych indeksowanych (numerycznych) oraz o stałych w programach zapisywanych w instrukcji biernej DATA (wraz z instr. READ i RESTORE).

Wykład ósmy: tablice, zmienne indeksowane (numeryczne) instrukcje: DIM, DATA, READ, RESTORE

W poprzednich wykładach wszystkie programy napisane były przy zastosowaniu zmiennych prostych (numerycznych lub alfanumerycznych). Zmienne proste nie pozwalają na zapamiętanie w programie dużej liczby wartości, a następnie wykonanie tych samych sekwencji instrukcji dla różnych zmiennych. Wprowadza się więc grupowanie ciągu zmiennych w tablice, które są identyfikowane przez swoje nazwy. Dostęp do poszczególnych elementów tablicy następuje przez podanie nazwy tablicy i indeksów określających żądany element.

Tak więc poindeksowany zbiór danych tego samego typu nazywa się **tablicą**. Wartościami indeksów mogą być pojedyncze liczby całkowite (tablica nazywa się wtedy jednowymiarową), pary liczb całkowitych (tablica dwuwymiarowa) itd. Tablicą jako całość jest identyfikowana przez swą nazwę.

Element tablicy, odpowiadający konkretnemu układowi wartości identyfikujących go indeksów, nazywa się **zmienną indeksowaną**. Zmienną taką oznacza się symbolem złożonym z napisanych po sobie: nazwy tablicy i ujętych w nawiasy (bądź rozdzielonych przecinkami) indeksami.

W języku BASIC należy określić nazwę i rozmiar tablic, jakie będą używane w danym programie. Do tego celu służy deklaracja tablicy.

Deklaracja tablicy w BASIC-u ZX Spectrum jest instrukcją biernej postaci:

n DIM nazwa tablicy (rozmiar)

gdzie:

n — numer wiersza

nazwa tablicy — składa się z:

- jednej litery dla tablic numerycznych
- jednej litery i znaku \$ dla tablic alfanumerycznych

rozmiar — zależy od liczby wymiarów (i elementów) tablicy
Tablice mogą być jednowymiarowe (nazywane wektorami), dwuwymiarowe (nazywane macierzami) lub trzywymiarowe itd.

Instrukcja DIM może występować w dowolnym miejscu programu (ale przed odwołaniem się do tej tablicy). Jeżeli w programie zadeklarowano kilka tablic o tej samej nazwie, to w chwili uruchomienia aktualna jest deklaracja w linii o najwyższym numerze. W tym samym programie może występować zmienna prosta numeryczna o tej samej nazwie co tablica. W trakcie rezerwowania miejsca dla danej tablicy numerycznej następuje nadawanie jej elementów wartości początkowych zero.

W wykładzie tym omawiać będziemy tylko tablice numeryczne, a tym samym zmienne indeksowe numeryczne. A teraz zilustrujemy przykładami to, co zostało już powiedziane.

Przykład 7.1

Dany jest wektor x o 10-elementach. Wyznaczyć sumę elementów o wartościach większych od zadanej stałej c.

```

10 REM Przykład 7.1
15 REM wektor definiujemy jako
   tablice jednowymiarowa,
   10 elementowa
20 DIM x(10)
25 INPUT "stała c = ";c
30 REM wczytanie wartości elem
   tow tablicy x
40 PRINT TAB(10);"TABLICA X"
45 PRINT
50 FOR i=1 TO 10
60 INPUT "podaj wartość x(i) =
";x(i)

```

```

70 PRINT x(i);" ";
80 NEXT i
85 LET s=0
90 FOR i=1 TO 10
100 IF x(i)>c THEN LET s=s+x(i)
110 NEXT i
115 PRINT
120 PRINT "Suma elementów o wa
rtosciach większych od zadane
j stałej c = ";c;" wynosi: ";
s
130 STOP

```

b) wyniki dla przykładowych danych

TABLICA X

1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10

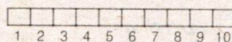
Suma elementów o wartościach większych od zadanej stałej c = 5 wynosi: 40

Instrukcja

20 DIM x(10)

deklaruje tablicę jednowymiarową o nazwie x i o 10 — elementach (liczba 10 deklaruje rozmiar, w tym wypadku liczbą elementów tablicy).

Tablicę jednowymiarową (wektor) można przedstawić w postaci wiersza (lub kolumny), tak więc dla naszej tablicy x w postaci



liczby 1 ÷ 10 kolejne liczby naturalne będące indeksami elementów (i — indeks; i = 1, 2, ..., 10)

x(i) — zmienna indeksowana numeryczna — i-ty element tablicy.

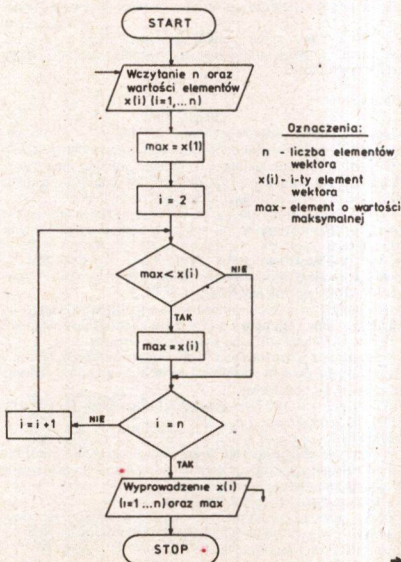
Instrukcja

60 INPUT „podaj wartość x (i) = „; x(i) nadaje wartość i-temu elementowi tablicy

Przykład 7.2

Napisać program znajdujący wartość maksymalną w wektorze n-elementowym

a) schemat blokowy



Oznaczenia:

- n - liczba elementów wektora
- x(i) - i-ty element wektora
- max - element o wartości maksymalnej

Oznaczenia:

n — liczba elementów wektora
 x(i) — i-ty element wektora
 max — element o wartości maksymalnej
 b) program.

```

10 REM Przykład 7.2
15 REM n - liczba elementów wektora
   i - indeks
   x(i) - i-ty element wektora
   (zmieniana indeksowana)
   max - wybrany element o wartości maksymalnej
20 INPUT "wprowadz n :";n
25 REM liczba elementów tej tablicy zostaje ustalona po wczycaniu wartosci n
30 DIM x(n)
35 REM wczytanie wartosci elementów tablicy x
40 FOR i=1 TO n
45 INPUT "wprowadz x(i):";x(i)
50 NEXT i
60 LET max=x(1)
70 FOR i=2 TO n
80 IF max<x(i) THEN LET max=x(i)
90 NEXT i
100 PRINT TAB(5);"wektor dany"
105 PRINT
110 FOR i=1 TO n
120 PRINT x(i); " ";
130 NEXT i
135 PRINT
140 PRINT "wartosc maksymalna :";max
150 STOP
    
```

Wykres dla przykładowych danych

wektor dany :

8 5 0 6 9 11 3 22

wartosc maksymalna : 22

d) uwaga.

Jeżeli chcielibyśmy znaleźć wartość minimalną w wektorze n-elementowym, wystarczy instrukcję o numerze 80 zapisać w postaci

```
80 IF max > x(i) THEN LET max = x(i)
```

oraz zamiast słowa maksymalnej (wartości) wpisła słowo minimalnej wartości.

Instrukcja

```
30 DIM x(n)
```

deklaruje tablicę jednowymiarową o nazwie x i o n-elementach. Zarezerwowanie odpowiedniego miejsca dla tej tablicy w pamięci komputera następuje po nadaniu wartości zmiennej n. Nadawanie tej wartości może nastąpić za pomocą instrukcji INPUT lub LET i musi nastąpić przed deklaracją tablicy.

Tablice jedno-, dwu- lub więcej wymiarowe zadeklarowane w powyższy sposób nazywa się często tablicami dynamicznymi. Odpowiednie instrukcje

```
INPUT "wprowadz x (i) :"; x(i)
```

powodują nadanie aktualnej wartości elementom tych tablic. Wartości te „pozostają” tak długo aż:

a) nie nadamy im nowych wartości instrukcją INPUT lub LET
 b) nie usuniemy programu z pamięci komputera, wykonując komendę NEW lub LOAD, lub wyłączymy komputer

Korzystając z „pozostawiania” tych wartości w tablicy, możemy odwoływać się do nich w różnych miejscach programu i tak np. odwołujemy się dodatkowo:

W instrukcji o nr. 100 w przykładzie 7.1 oraz w instrukcjach o nr. 80 i 120 w przykładzie 7.2

Przykład 7.3

W ciągu pewnego czasu dla grupy 15 uczniów przeprowadzono 5 sprawdzianów z wybranego przedmiotu. Obliczyć średnią ocenę otrzymaną przez każdego ucznia.

a) analiza zadania

Ocenę każdego ucznia możemy zapisać w sektorze jednowymiarowym pięcioelementowym, ale tych wektorów mielibyśmy 15. Zapiszemy oceny w postaci tablicy.

```

10 REM Przykład 7.3
20 DIM o(15,5)
25 REM wartosci elementow tablicy wprowadzamy wierszami
   i=1 TO 15
   j=1 TO 5
40 INPUT "wprowadz o(i,j): ";o(i,j)
45 NEXT j
50 NEXT i
60 REM dla kazdego i obliczamy nowa sume - sr=0
70 LET sr=0
75 PRINT "uczen ";i;"oceny : "
80 FOR j=1 TO 5
90 LET sr=sr+o(i,j)
95 PRINT o(i,j); " ";
100 NEXT j
105 LET sr=sr/5
110 PRINT "ocena srednia : ";sr
115 PRINT
120 NEXT i
130 STOP
    
```

gdzie: i-ty wiersz — i-ty uczeń

j-ta kolumna — j-ta ocena

o_{ij} — oznacza j-tą ocenę i-tego ucznia ($i=1..15$,

$j=1,2,3,4,5$)

Jest to tablica dwuwymiarowa (macierz), którą deklarujemy w następujący sposób:

```
20 DIM o(15,5)
```

Każdy element tej tablicy jest opisany dwójką liczb naturalnych będących jego indeksami i tak:

$o(1,2)$ — jest to element pierwszej wiersza i drugiej kolumny

$o(5,3)$ — jest to element piątej wiersza i trzeciej kolumny

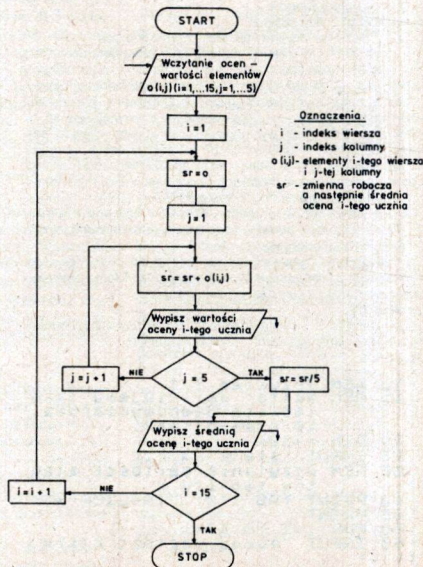
$o(i,j)$ — jest to element i-tego wiersza i j-tej kolumny

Tak więc pierwszy indeks oznacza numer wiersza, a indeks drugi — numer kolumny.

Ocenę średnie obliczamy wg wzoru:

$$sr = \frac{\sum_{j=1}^5 o_{ij}}{5} \quad \text{dla każdego } i \text{ (dla każdego ucznia) } i = 1..15$$

b) schemat blokowy



Oznaczenia:

i - indeks wiersza
 j - indeks kolumny
 $o(i,j)$ - elementy i-tego wiersza i j-tej kolumny
 sr - zmienna robocza a następnie średnia ocena i-tego ucznia

Oznaczenia:

- i — indeks wiersza
- j — indeks kolumny
- $0(i, j)$ — element i-tego wiersza i j-tej kolumny
- sr — zmienna robocza, a następnie średnia ocena i-tego ucznia
- c) program

Przykład 7.4

Znaleźć sumę oraz liczbę elementów dodatnich leżących powyżej głównej przekątnej macierzy Z ($z_{ij}, i, j = 1, 2, \dots, n$)

```

10 REM Przykład 7.4
15 REM n — liczba wierszy (kolumn) macierzy z
   i — indeks wiersza
   j — indeks kolumny
   z(i, j) — element macierzy z
   s — suma elementów dodatnich dla których i > j
   l — liczba elementów dodatnich, dla których i > j
20 INPUT "wprowadz n : "; n
REM liczba elementów tej tablicy zostaje ustalona po wczytaniu n
30 DIM z(n, n)
REM wartości elementów tablicy wprowadzamy wierszami
40 PRINT "macierz dana : "; PR
INT
45 FOR i=1 TO n
50 FOR j=1 TO n
60 INPUT "wprowadz z(i, j) : "; z(i, j)
65 PRINT z(i, j) " "
70 NEXT j
75 LET s=0: LET l=0
80 FOR j=i+1 TO n
85 IF z(i, j) > 0 THEN LET s=s+z(i, j)
95 NEXT j
100 NEXT i
105 PRINT "liczba elementów dodatnich powyżej głównej przekątnej : "; l: "ich sum s = "; s
110 STOP

```

a) analiza zadania

Zapiszemy macierz Z w postaci

$$Z = \begin{pmatrix} z_{11} & z_{12} & \dots & z_{1n} \\ z_{21} & z_{22} & \dots & z_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n1} & z_{n2} & \dots & z_{nn} \end{pmatrix}$$

Jeżeli liczba wierszy danej macierzy równa się liczbie jej kolumn, to macierz nazywa się kwadratową, jeżeli nie — prostokątną. Położenie elementu $z(i, j)$ w macierzy Z można określić w następujący sposób:

- $i > j$ znajduje się powyżej głównej przekątnej
- $i = j$ znajduje się na głównej przekątnej
- $i < j$ znajduje się poniżej głównej przekątnej

Badanie znaku i położenia elementów $z(i, j)$ będziemy przeprowadzać w kolejnych wierszach macierzy. Macierz Z zostanie zadeklarowana jako dynamiczna tablica dwuwymiarowa (n x n)-elementowa.

Narysowanie schematu blokowego pozostawiamy czytelnikowi. b) program.

Instrukcja

30 DIM a(15,5)

deklaruje tablicę dwuwymiarową o nazwie a i liczbie elementów równej 75 (15x5).

Kolumny Wiersze	1	2	3	4	5
1	0 _{1,1}	0 _{1,2}	0 _{1,3}	0 _{1,4}	0 _{1,5}
2	0 _{2,1}	0 _{2,2}	0 _{2,3}	0 _{2,4}	0 _{2,5}
⋮	⋮	⋮	⋮	⋮	⋮
15	0 _{15,1}	0 _{15,2}	0 _{15,3}	0 _{15,4}	0 _{15,5}

natomiast instrukcja

30 DIM z(n,n)

deklaruje dynamiczną tablicę dwuwymiarową o nazwie z i liczbie elementów równej n x n, liczba tych elementów jest znana w momencie nadania wartości zmiennej n.

Każda tablica jest przedstawiona w pamięci komputera jako wektor. Tak więc kolejna tablica dwuwymiarowa jest przedstawiona jako ciąg kolejnych kolumn tworzących wektor o liczbie elementów równej liczbie elementów danej tablicy. A dostęp do tego wektora musi się odbywać zawsze tak jak do tablicy dwuwymiarowej.

Tablicą dwuwymiarową jest również tablica złożona z jednego wiersza (x) o 15 elementach lub kolumny (z) o 12 elementach, zadeklarowana w następujący sposób:

10 DIM x(1,15)

lub

20 DIM z(12,1)

BASIC na ZX Spectrum nie dopuszcza napisania deklaracji tablic w następujący sposób:

10 DIM x(1,15), z(12,1)

Przed każdą nazwą tablicy należy napisać słowo DIM. Tak więc deklarację powyższych tablic możemy napisać też w postaci:

10 DIM x(1,15) : DIM z(12,1)

Przykład 7.5.

Dany jest n-elementowy wektor w ($w_i, i=1, 2, \dots, n$). Odwrócić kolejność elementów wektora wg wzoru.

$$\left. \begin{aligned} W_i &= W_{n-i+1} \\ W_{n-i+1} &= W_i \end{aligned} \right\} i = 1, 2, \dots, [i]$$

([X] — część całkowita x)

a) program.

```

10 REM Przykład 7.5
15 REM n — liczba elementów wektora
   i — indeks
   w(i) — i-ty element wektora
20 INPUT "wprowadz n : "; n
REM wektor deklarujemy jako tablicę jednowymiarową, n-elementową
30 DIM w(n)
35 PRINT "wektor dany : "; PR
INT
40 FOR i=1 TO n
45 INPUT "wprowadz w(i) : "; w(i)
50 PRINT w(i) " "
55 NEXT i: PRINT
60 LET k=INT (n/2)
65 REM r — zmienna robocza, wykorzystywana do "przechowywania" wartości w(i) przy przestawianiu
70 FOR i=1 TO k
75 LET r=w(i)

```

```

80 LET w(i)=w(n-i+1)
85 LET w(n-i+1)=r
90 NEXT i
95 PRINT "nowy wektor : "; PR
INT
100 FOR i=1 TO n
105 PRINT w(i); " ";
110 NEXT i
115 STOP

```

b) wyniki dla przykładowych danych

```

wektor danu :
2 4 6 8 10 12 14 16 18
nowy wektor :
18 16 14 12 10 8 6 4 2

```

Przykład 7.6.

Zaprogramować w BASICU:

1) wypełnienie tablicy A ($a_i, i = 1, 2, \dots, n$) liczbami naturalnymi tzn.

$$a_1 = 1, a_2 = 2, \dots, a_n = n$$

2) obliczenie sumy:

$$s = \sum_{n=1}^n c \cdot a_i$$

gdzie: $c = -1$ dla a_i , podzielnych przez 3

$c = +1$ dla pozostałych

3) wpisanie liczb a_i podzielnych przez 3 do tablicy o nazwie B.

a) analiza zadania.

Elementom tablicy A będziemy nadawać wartości w trakcie obliczeń za pomocą instrukcji podstawienia. Do sprawdzenia czy dana wartość dzieli się przez 3, używamy funkcji standardowej INT (część całkowita liczby, może być ujemna; podaje najbliższą liczbę całkowitą mniejszą od argumentu). Jeżeli liczba dzieli się przez 3, to sumujemy ze znakiem — oraz wpisujemy tę wartość do tablicy B, jeżeli nie, to sumujemy ze znakiem +.

Narysowanie schematu blokowego pozostawiamy czytelnikowi.

b) program

```

10 REM Przykład 7.6
15 REM n - liczba naturalna
      (liczba elementów
      tablicy A)
      i - kolejna (liczba na-
      turalna) indeksa ta-
      blicy A)
      A(i) - element tablicy A
      B(j) - element tablicy B
      j - indeks tablicy B
      s - suma elementów
      w tablicy B)
      s - suma elementów ta-
      cuki A z odpowiednim
      znakiem)
20 INPUT "wprowadz n : "; n
30 DIM A(n); DIM B(n)
35 REM nie znamy liczby elemen-
      tow tablicy wyników,
      więc deklarujemy ją ja-
      ko tablicę o liczbie
      elementów = liczbie ele-
      mentów tablicy wejścio-
      wej
40 LET s=0; LET j=0
50 FOR i=1 TO n
60 LET A(i)=i
70 LET r=i/3
75 IF r>INT(r) THEN GO TO 90
80 LET j=j+1; LET E=-1
85 LET B(j)=i
90 LET s=s+E*A(i)
100 NEXT i
105 PRINT "tablica liczb natura-
lnych          od 1 do "; n
110 PRINT
115 FOR i=1 TO n
120 PRINT " "; A(i);
125 NEXT i; PRINT
130 PRINT "tablica liczb natur-
alnych        podzielnych prze

```

```

135 PRINT
140 FOR k=1 TO j
145 PRINT " "; B(k);
150 NEXT k; PRINT
155 PRINT TAB(8); "Suma = "; s
160 STOP

```

W programach z przykładów 7.5 i 7.6 zastosowano instrukcje INPUT i LET do nadawania wartości elementom tablic (zmiennym indeksowanym).

Ponadto w programach tych pokazano, iż na zmiennych indeksowanych numerycznych możemy wykonywać te same działania, jakie wykonujemy na zmiennych prostych numerycznych.

Z dotychczasowych wykładów wiemy, że aby wpisać w formie konserwacyjnej pewną liczbę wartości zmiennych, używamy instrukcji INPUT. Czasami zdarza się, że znamy te wartości przed wykonaniem programu, wtedy możemy je umieścić w programie dzięki instrukcji READ (czytaj) i DATA (dane). Wykonując program komputer sam je umieści w odpowiednim momencie w pamięci. Instrukcje te można też zastosować do nadawania wartości elementom tablic (zmiennych indeksowanych).

Przykład 7.7

Dany jest program:

```

10 REM Przykład 7.7
15 REM wypełnienie tablic x
      i w tym samym wartos-
      ciami
20 DIM x(3,5); DIM y(3,5)
25 REM nadawanie wartosci ele-
      mentom tablicy x - wier-
      szami
30 FOR i=1 TO 3
40 FOR j=1 TO 5
50 READ x(i,j)
60 NEXT j
70 NEXT i
80 RESTORE
90 REM nadawanie wartosci ele-
      mentom tablicy y - ko-
      lumnami
100 FOR j=1 TO 5
110 FOR i=1 TO 3
120 READ y(i,j)
130 NEXT i
140 NEXT j
150 PRINT TAB(5); "tablica x"
165 PRINT
170 FOR i=1 TO 3
180 FOR j=1 TO 5
190 PRINT " "; x(i,j);
200 NEXT j
205 PRINT
210 NEXT i
215 PRINT
220 PRINT TAB(5); "tablica y"
225 PRINT
230 FOR i=1 TO 3
240 FOR j=1 TO 5
250 PRINT " "; y(i,j);
260 NEXT j
265 PRINT
270 NEXT i
280 STOP
290 DATA 1,2,3,4,5,6,7,8,9,10
300 DATA 11,12,13,14,15

```

W programie tym zastosowano instrukcje READ, DATA i RESTORE. W trakcie wprowadzania linii programu jest możliwe tworzenie zestawu danych, którymi będzie operował program. Do tego celu służy instrukcja bierna DATA postaci:

-n DATA lista

gdzie:

n — numer wiersza

lista — dowolna liczba stałych numerycznych lub alfanumerycznych oddzielonych od siebie przecinkami.

Występujące w programie instrukcje DATA tworzą według rosnących numerów wierszy, jeden zestaw danych. I tak w naszym programie instrukcje DATA o numerach wierszy 230 i 240 tworzą jeden zestaw danych postaci:

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15

Instrukcje DATA można napisać w dowolnym miejscu programu. Zaleca się, aby instrukcje te pisać na końcu programu, grupując wszystkie z danego programu obok siebie.

Do przenoszenia wartości z zestawu DATA pod zmienne programu służy instrukcja czynna READ postaci:

n READ lista

gdzie: n — numer wiersza

lista — nazwy zmiennych prostych numerycznych lub alfanumerycznych, zmiennych indeksowanych numerycznych lub alfanumerycznych oddzielone od siebie przecinkami.

Wykonanie instrukcji READ powoduje pobieranie kolejnych wartości stąych z zestawu danych (poczynając od najniższego numeru wiersza instrukcji DATA) i podstawianie ich pod kolejne nazwy zmiennych występujące na liście READ. Jeżeli zestaw danych z pierwszej instrukcji DATA zostanie wyczerpany, to pobieranie następuje z zestawu danych następnej instrukcji DATA o wyższym numerze wiersza (pierwszym wyższym).

W tym programie wykonanie instrukcji READ w ciągu instrukcji postać:

```
30 FOR i=1 TO 5
40 FOR j=1 TO 3
50 READ x(i,j)
60 NEXT j
70 NEXT i
```

spowoduje wyczerpanie wszystkich danych z zestawu danych instrukcji DATA „zabrakto” więc danych do wykonania ciągu instrukcji o numerach 100—140 (brak DATA o wyższych numerach wierszy). Wykonanie instrukcji RESTORE powoduje przydzielenie do aktualnego czytania zestawu danych numerycznych w instrukcji DATA (230 i 240) poczynając od najniższego numeru wiersza instrukcji DATA (tego samego zestawu danych).

Instrukcja RESTORE ma postać:
n RESTORE

gdzie: n — numer wiersza

Instrukcji RESTORE używa się wtedy, gdy trzeba wykonać obliczenia na tym samym zestawie danych (umieszczonych w DATA) lub przyporządkowania tych samych wartości kilku zmiennym w kilku miejscach tego samego programu. Wykonanie tej instrukcji powoduje potraktowanie kolejnej instrukcji READ tak, jakby była pierwszą instrukcją READ w programie, tzn. będzie ona pobierać wartości z pierwszej instrukcji DATA w danym programie (instrukcja o najniższym numerze wiersza).

Przykład 7.8.

Dane jest 8 wektorów 5-elementowych o znanych wartościach elementów. Znaleźć liczbę wektorów, których długość nie przekracza zadanej wielkości d.

a) analiza zadania

Długość wektora x obliczamy według wzoru:

$$l = \sqrt{x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2}$$

```
10 REM Przykład 7.8. Kwadratowa x(i)
l - długość wektora
d - zadana wielkość
m - liczba wektorów
o - długość <= d
```

```
20 INPUT "wprowadz d :";d
30 DIM x(5)
35 LET m=0
40 FOR i=1 TO 8
45 PRINT "wektor danu : "
50 LET s=0
55 FOR j=1 TO 5
60 READ x(j)
65 PRINT " ";x(j);
70 LET s=s+x(j)*x(j)
75 NEXT j
80 LET l=SQR(s)
90 IF l<=d THEN LET m=m+1
95 PRINT
100 PRINT "dlugosc wektora : ";
105 NEXT i
110 PRINT "liczba wektorow, ktoru
rych dlugosc nie przekracza zada
nej wielkosci - ci d =";d;" wyn
osi :";m
115 STOP
120 DATA 1,3,5,7,9,2,4,6,8,10
125 DATA 1,0,0,5,8,9,3,4,6,7,10
130 DATA 1,0,0,3,4,5,6,7,8,9,10
135 DATA 10,1,9,2,6,8,3,7,4,6,5
```

Wartość obliczeń wektorów znany przed wykonaniem programu, więc możemy je umieścić w programie stosując instrukcję DATA. Nadawcą będąmy wartości elementów pierwszego wektora, oblicząc jego długość i porównując z wielkością d, potem drugiemu, trzeciemu wektorowi itd.

Narysowanie schematu blokowego pozostawiamy czytelnikowi.

b) program

Jeżeli chcieliśmy programy z przykładów 7.7 i 7.8 wykonać dla innego zestawu danych, należałoby zmienić listę stąych (numerycznych) w instrukcji DATA. Zmiany tej dokonujemy przed wykonaniem programu.

Tak więc, jeżeli program ma wykonywać obliczenia dla różnych partii danych, należy zastosować instrukcję INPUT.

W instrukcji DATA umieszcza się występujące zazwyczaj w programie stałe, z których korzystamy kilka razy w danym programie.

Zadania:

Napisać programy w BASIC-u realizujące następujące zadanie:

1. Dane są macierze prostokątne A.B.C (i = 1,2,...,m, j = 1,2,...,n) Obliczyć macierz D według wzoru:

$$D = p \cdot (A + B) + C$$

gdzie: p jest zadana liczbą (skalarem)

2. Dana jest macierz kwadratowa A(a_{ij}, i,j = 1,2,...,n)

Obliczyć sumę tych elementów, które należą do danego przedziału [c,d]

3. Dane jest n-vektorów k-elementowych. Znaleźć liczbę wektorów, których długość nie przekracza zadanej wielkości d.

Uwaga: jest to uogólnione zadanie z przykładu 7.8. Wektory należy umieścić w tablicy dwuwymiarowej, w której wiersz reprezentuje wektor k-elementowy (porównaj przykład 7.3).

ZZW

BADANIE KIEROWCÓW

Mieczysław SKONIECZNY

Ten program służy do badania orientacji przestrzennej i sybkości podejmowania decyzji przez kierowcę. Pierwotorem prezentowanego symulatora mikrokomputerowego jest aparat krzyżowy DUFOUR.

Prezentowany program mikrokomputerowy, eksploatowany na mikrokomputerze typu AMSTRAD CPC 6128, składa się z dwóch etapów:

— próbnego — złożonego z 4 impulsów. Jego celem jest przygotowanie kierowcy do badania właściwego oraz oswajanie go ze sprzętem mikrokomputerowym;

— właściwego — złożonego z 10 impulsów, gdzie przy każdej emisji liczony jest przez komputer czas wykonania zadania.

Jeden impuls polega na wyświetleniu na ekranie monitora dwóch obiektów:

— statycznego (kwadratowa ramka)

— dynamicznego (sterowanego przez badanego) — wypelniony kwadrat.

Zadaniem badanego jest umieszczenie obiektu dynamicznego (kwadratu) w obiekcie statycznym (ramce). Sterowanie kwadratem odbywa się za pomocą czterech klawiszy funkcyjnych, rozmieszczonych z prawej strony klawiatury.

↑ — ruch do góry

↓ — ruch w dół

→ — ruch w prawo

← — ruch w lewo

Rozmieszczenie obiektów na ekranie jest generowane losowo przez mikrokomputer.

Aby zapewnić porównywalność impulsów, odległość obu obiektów w momencie startowym jest stała. Po pomyślnym zareagowaniu na impuls badany kierowca słyszy efekt dźwiękowy i na dole ekranu wyświetlany jest czas jego reakcji. Na zakończenie programu podawany jest średni czas reakcji i ocena końcowa w skali czterostopniowej. Dodatkowo pojawia się ocena słowna, określająca stopień przydatności do wykonywania zawodu kierowcy.

Kryteria graniczne poszczególnych ocen w skali czterostopniowej wyznaczono metodami statystycznym. Zbadano próbki losową 150 kierowców wojskowych i uzyskane przez nich wyniki poddano analizie statystycznej, uwzględniającej charakterystyki rozkładu normalnego.

UWAGI PROGRAMOWE

1. Jeżeli użytkownik chciałby zmienić graficzną reprezentację obiektu dynamicznego i statycznego, może uczynić to w prosy sposób. W tym celu wystarczy usunąć podprogram z linii 640, który rysuje na ekranie kwadrat i ramkę. Zamieści program proponujemy wprowadzić ciąg instrukcji:
640 LOCATE V - 1, X - 1: PRINT CHR\$(144), CHR\$(145), CHR\$(146)
642 LOCATE V - 1, X: PRINT CHR\$(147), CHR\$(148)
644 LOCATE V - 1, X + 1: PRINT CHR\$(149), CHR\$(150), CHR\$(151)
646 RETURN

Efektom działania tego podprogramu jest wyrzutowanie w określone (współrzędne x,y) miejsce ekranu obiektu statycznego. Dodatkowo należy zwiększyć zestaw symboli graficznych, definiowanych przez użytkownika, zastępując instrukcję 700 ciągami instrukcji:

```
700 SYMBOL AFTER 143
701 SYMBOL 143,60,126,255,255,255,126,60
702 SYMBOL 144,0,0,1,7,15,30
703 SYMBOL 145,0,0,0,255,255,129
704 SYMBOL 146,0,0,0,128,224,240,240,120
705 SYMBOL 147,30,28,28,28,28,28,30
706 SYMBOL 148,120,56,56,56,56,56,120
707 SYMBOL 149,30,15,15,7,1
708 SYMBOL 150,0,129,255,255,255
709 SYMBOL 151,120,240,240,224,128
```

Aby wprowadzone zmiany nie kolidowały z dotychczasową konstrukcją programu, należy usunąć instrukcję 620. Gdyby użytkownik chciał stosować inne obiekty dynamiczne, musi dokonać analogicznych modyfikacji programu.

2. W etapie właściwym po każdej poprawnej reakcji na impuls w dowolnej części ekranu wyświetlana jest informacja o czasie reakcji badanego kierowcy. Informację tę można uzupełnić podając dodatkowo dotychczasowy średni czas reakcji oraz komentarz oceniający dotychczasową działalność badanego. W tym celu należy instrukcję 660 zastąpić następującymi instrukcjami:

```
10 CLEAR INPUT: BORDER 0: PAPER 0: CLS: MODE 1: INK 0,0
   INK 2,24: GOSUB 960
20 WINDOW #1,6,32,5,12: PAPER #1,2: CLS #1: PEN #1,0
LOCATE #1,1,2: PRINT "BADANIE", LOCATE #1,2,4: PRINT
"ORGANIZACJA PRZESTRZENNA", LOCATE #1,2,5: PRINT
"NTA1", "SI ZYWKOCI PODJEJMONIA", LOCATE #1,2,6: PRINT
"NTA1", "RECYZJA PRZECI KIEROWCO": INK 2,24,26
```

```
30 IF INKEY="" THEN GOTO 30
40 MODE 1: INK 1,14: INK 2,24: INK 3,6: WINDOW #1,1,40
   1,7: PAPER #1,2: CLS #1: PEN #1,0: LOCATE #1,2,2: PRINT
   "PROGRAM ZA ZADANIE ZBADA" ORGANIZA": LOCATE
   #1,2,3: PRINT "NTA1", "CJ PRZESTRZENNA I ZYWKOCI"
   PODEJMONIA: GOTO 30
```

```
50 LOCATE #1,2,4: PRINT "NTA1", "NIA DECZYLI W BADANEGO U
   FYSYKONIE KIE": LOCATE #1,2,5: PRINT "NTA1", "ROWCY WNOJE
   KOWEGO. SKLADA ON SI Z": LOCATE #1,2,6: PRINT "NTA1",
   "WU PRZEBIEGU I WIAZALIEGO."
```

```
60 WINDOW #2,2,39,9,15: PAPER #2,1: CLS #2: PEN #2,0
LOCATE #2,2,2: PRINT "ETAP PRZEBIEGU SI ZLOZYLO Z
   4 PR50 NA": LOCATE #2,2,3: PRINT "NTA2", "I JICH NA CELU
   PRZYKOTOWANIE BADANEGO": LOCATE #2,2,4: PRINT "NTA2",
   "DO ETAPU WIAZALIEGO I OSWOJENIE NA"
```

```
70 LOCATE #2,2,5: PRINT "NTA2", "SPRITEN MIKROKONTROW
   WY. NIE JEST": LOCATE #2,2,6: PRINT "NTA2", "BADANY CZAS
   REAKCJI."
```

```
80 WINDOW #3,2,39,17,22: PAPER #3,5: PEN #3,0: CLS #3
LOCATE #3,2,2: PRINT "NTA3", "ETAP WIAZALIEGO SKLADA SI Z
   10 PR50": LOCATE #3,2,3: PRINT "NTA3", "GDZIE NA KAJEDZ
   Z NICH JEST NALICZANY": LOCATE #3,2,4: PRINT "NTA3", "PRZECI
   MIKROKOMPUTER CZAS REAKCJI NA"
```

```
90 PRINT "NTA3,5: PRINT "NTA3", "DANEGO." PEN 2: LOCATE 1,
   25: PRINT "MACIOWI" "SPACZO"
```

```
100 IF INKEY="" THEN 100
110 MODE 1: WINDOW #1,1,40,1,7: PAPER #1,2: CLS #1: PEN #1,0
LOCATE #1,1,40,1,7: PRINT "NTA1", "JEDNA PR50A POLEGA NA
   WYMIENIENIU NA": LOCATE #1,2,3: PRINT "NTA1", "EKRANIE
   MONITORA 2 OBIKT: S": LOCATE #1,3,4: PRINT "NTA1", "- STA
   TYCZNEGO (KWADRATA RAJKA)"
```

```
120 LOCATE #1,3,5: PRINT "NTA1", "DYNAMICZNE (STEROWANE
   KIEROWCE) PRZECI NA": LOCATE #1,5,6: PRINT "NTA1", "DANEGO"
   W WYPELNIENIU KWADRAT."
```

```
130 WINDOW #2,1,40,9,16: PAPER #2,1: PEN #2,0: CLS #2
LOCATE #2,2,2: PRINT "NTA2", "ZADANIE KIEROWCY JEST WY
   ESCZANIE": LOCATE #2,2,3: PRINT "NTA2", "OBIEKT DYNAMICZ
   NEGO (KWADRAT) W OBIE": LOCATE #2,2,4: PRINT "NTA2", "KIE
   I STATYCZNEJ (RAJKA)."
```

```
140 LOCATE #2,2,5: PRINT "NTA2", "STEROWANIE KWADRATEM
   BYWA SI Z NA PO": LOCATE #2,2,6: PRINT "NTA2", "WOCII KLA
   WISZY FUNKCYJNYCH ROZAJE": LOCATE #2,2,7: PRINT "NTA2",
   "SZCZONYCH Z PRAWY STRONY KLAWIATURY."
```

```
150 PEN 3: LOCATE 4,18: PRINT CHR$(240): " - RUCH DO
   GORY": LOCATE 4,20: PRINT CHR$(241): " - RUCH NA DOL"
LOCATE 4,22: PRINT CHR$(243): " - RUCH W PRAWO": LOCATE
   4,24: PRINT CHR$(242): " - RUCH W LEWO": PEN 1: PRINT
   INT " *": PRINT "MACIOWI" "SPACZO"
```

```
160 IF INKEY="" THEN 160
170 MODE 1: WINDOW #1,1,40,2,12: PAPER #1,1: CLS #1: PEN #1,0
LOCATE #1,1,2: PRINT "NTA1", "ROZACIECZANIE OBIKTW NA
   EKRANIE JEST GOMEROWANE LOSOWO PRZECI ALIKROPUTY
   TER. JEDNAK, POPY ZAPAMIENI POR50WALNIO": PR50DIE
   QIOT": OBIKTW W ANONCIE STAR."
```

```
180 LOCATE #1,1,6: PRINT "NTA1", "TOWYMA JEST STALA." LOCATE
   #1,1,8: PRINT "NTA1", "PO POWY7YNA ZAKOJECIENIA PR50BY
   BADANY SLYSZY EFEKT DWIJKOWY I W DOLU EKRANU J
   EST WYMIENIENIU CZAS REAKCJI."
```

```
190 WINDOW #2,2,40,15,20: PAPER #2,2: CLS #2: PEN #2,0
LOCATE #2,2,2: PRINT "NTA2", "NA ZAKOJECIENIE PROGRAMU JEST
   PODAWANY": LOCATE #2,2,3: PRINT "NTA2", "CZAS REAKCJI I
   CENA BADANEGO W SKALI": LOCATE #2,2,4: PRINT "NTA2", "CZTER
   OSTOPIOWYCH." PEN 2: LOCATE 1,24: PRINT "MACIOWI" "SP
   ACZO"
```

```
200 IF INKEY="" THEN 200
210 MODE 0: BORDER 14: PAPER #1: CLS #1: WINDOW #1,3,18,12
16: PAPER #1,3: CLS #1: PEN #1,0: LOCATE #1,3,3: PRINT "NTA1",
   "ETAP PRZEBIEGU": INK 0,0,24
220 IF INKEY="" THEN 220
230 MODE 1: INK 0,14: INK 1,0: INK 2,1: INK 3,24: BORDER
```

```
656 MODE 1: PEN 2: G1 = TIME: CZ = CZ + (G1 - G) / 300
LOCATE #1
657 PRINT "CZAS REAKCJI NA IMPULS NR": R
658 PEN 3: LOCATE 18,2: PRINT (G1 - G) / 300, "GEK"
659 PEN 2: LOCATE 5,4: PRINT "DOTYCHCZASOWY SREDNI
   CZAS REAKCJI"
660 PEN 3: LOCATE 15,25: PRINT CZ / R, "SEK"
661 IF R = 1 THEN GOTO 670
662 IF (G1 - G) / 300 < CZ / R THEN GOTO 665
663 LOCATE 17: PRINT "TEMPO SLABNIE - POPRAW
   ZYWKOSC REAKCJI"
664 GOTO 670
665 LOCATE 4,7: PRINT "JESTEŚ DOBRY. UTRZYMAJ TO
   TEMPO"
670 AS = INKEY$
680 IF INKEY$ = " " THEN 680
```

Dodatkowo należy usunąć instrukcje o etykietce 220.
3. Stosunkowo prostymi modyfikacjami programowymi można uzyskać sterowanie obiektem dynamicznym za pomocą innych klawiszy. W tym celu należy dołączyć do programu źródłowego następujący podprogram:

```
1002 MODE 1: LOCATE 1,24: INPUT "CZY CHCESZ ZDEFINIOWAC
   WLASNE KLAWISZE STERUJACE (T/N)": KS
1005 IF KS = "T" OR KS = "t" THEN GOTO 1020
1010 IF KS = "n" OR KS = "N" THEN GOTO 1015
1015 LOCATE 1,25: PRINT "GOTO 1000"
1020 LOCATE 1,1: INPUT "RUCH DO GORY" - BS (1)
1025 LOCATE 1,3: INPUT "RUCH NA DOL" - BS (2)
1030 LOCATE 1,5: INPUT "RUCH W PRAWO" - BS (3)
1035 LOCATE 1,7: INPUT "RUCH W LEWO" - BS (4)
1040 RETURN
1050 BS (1) = CHR$(240)
1052 BS (2) = CHR$(241)
1054 BS (3) = CHR$(242)
1056 BS (4) = CHR$(243)
1058 GOTO 1040
```

Dodatkowo należy wprowadzić instrukcję 39 GOSUB 1002 oraz zmodyfikować instrukcję 410-440 w następujący sposób:

```
410 IF AS = BS (1) GOTO 450
420 IF AS = BS (2) GOTO 490
430 IF AS = BS (3) GOTO 530
440 IF AS = BS (4) GOTO 570
```

Mieczysław SKONIECZNY

```
R 14: PAPER 0: P1=0: CLS
240 FOR R=1 TO 5
250 GOSUB 520
260 FOR I=1 TO 1000: NEXT
270 NEXT
280 P1=1: INK 1,6
```

```
290 MODE 0: INK 1,14: INK 2,24: INK 3,6: INK 0,0: BORDER
R 14: PAPER 1: CLS: WINDOW #1,3,18,12,16: PAPER #1,3: CLS
L50: PEN #1,0: LOCATE #1,2,3: PRINT "NTA1", "ETAP PRZEBIEGU
   WIAZALIEGO I OSWOJENIE NA"
```

```
300 IF 0,074: INKEY="" THEN 300
310 FOR H=1 TO 2000: NEXT
320 FOR R=1 TO 10
340 GOSUB 520
350 CJ=CZ+(61-61/200)
```

```
360 NEXT
370 FOR H=1 TO 2000: NEXT
380 GOSUB 960: MODE 1: INK 0,0: INK 1,0: INK 2,24: INK
   3,6: BORDER 24: PAPER 2: CLS: WINDOW #1,3,17,2,4: PAPER
   #1,0: CLS #1: PEN #1,2: LOCATE #1,2,2: PRINT "NTA1", "WYNIK
   I TESTU"
```

```
390 PEN 1: LOCATE 1,7: PRINT "SREDNI CZAS REAKCJI - "
   ": PRINT USING "###.##": CJ/10: PRINT "SEK"
400 PEN 3: LOCATE 9,12: PRINT "##### OCENA #####": PEN 1
```

```
410 CJ=CZ/10
420 IF CJ<4,74 GOTO 490
430 IF CJ<5,57 GOTO 490
440 IF CJ<6 GOTO 470
450 GOTO 660
```

```
440 LOCATE 1,14: PRINT "TOWYMA WYNIKI SI BARDZO SZARJE
   - BADANIE ORGANIZACJI PRZESTRZENNEJ I ZYWKOCI
   PODEJMONIA RECYZJA DYSKALIFIKUJE CIOTKIA JAKIE
   ROWNO WJOSKOWEGO. RECYZJA ZYDZEWYMANIE WIAZALIEGO": PR
   INT "NTA3,5: PRINT "MACIOWI" "SPACZO"
470 LOCATE 1,14: PRINT "TOWYMA WYNIKI SI SZARJE. BADAN
   IE ORGANIZA-CJI PRZESTRZENNEJ I ZYWKOCI PODEJMONIA
```

```

A-NIA DECYZJI KWALIFIKUJE CIO JAKO KIE- ROMCO WD
JKSONEGO.ALE MUSISZ ZDECYDOWANIEJWCZY).:PRINT:PE
N 3:PRINT"UZYKAZEB OCENO 3":GOTO 500
480 LOCATE 1,14:PRINT"MOJE WYNIKI SĄ Dobre. BADA
TE ORGANIZACJA PRZESTRZENNEJ I SYBKOCCI PODEJMOW
A-NIA DECYZJI KWALIFIKUJE CIO JAKO KIE- ROMCO WD
JKSONEGO.WIE JAKOJEDY JEDNAK JAK BODLIEZ JWCZY
V2.:PRINT:PEN 3:PRINT"UZYKAZEB OCENO 4":GOTO 50
0
490 LOCATE 1,14:PRINT"MOJE WYNIKI SĄ BARDZO Dobre
- BADAНИЕ ORGANIZACJI PRZESTRZENNEJ I SYBKOCCI
PODEJMOWANIA DECYZJI KWALIFIKUJE CIO JAKO KIE
ROMCO WDJKSONEGO.WIE MUSISZ JWCZY).:PRINT:PE
N 3:PRINT"UZYKAZEB OCENO 5":GOTO 500
500 IF NOT(INKEY="s" OR INKEY="S") THEN 500
510 GOTO 10
520 NODE 1:INK 1,4;INK 1,0;INK 2,1;INK 3,26;BORDE
R 14:PAPER 0:CLS
530 PEN 1
540 v=2+INT(22*HRND)
550 v=2+INT(37*HRND)
560 IF v>12 GOTO 580
570 v=(v+2):GOTO 510
580 x10=-12
590 IF v>20 GOTO 610
600 IF v>15:GOTO 620
610 y10=y-1
620 CLEAR INPUT:PEN 1:GRAPHICS PEN 3:GOSUB 890:LOC
ATE y1,x1:PRINT CHR$(143);
630 q=TIME
640 a$=INKEY$
650 IF a$="" THEN GOTO 440
660 IF a$=CHR$(240) GOTO 700
670 IF a$=CHR$(241) GOTO 740
680 IF a$=CHR$(242) GOTO 780
690 IF a$=CHR$(243) GOTO 820

```

```

700 IF x1=1 GOTO 640
710 FRAME:LOCATE y1,x1:PRINT CHR$(128);:LOCATE y1,
x1-1:PRINT CHR$(143);:y1=y1-1
720 FRAME
730 GOTO 860
740 IF x1=25 GOTO 640
750 FRAME:LOCATE y1,x1:PRINT CHR$(128);:LOCATE y1,
x1-1:PRINT CHR$(143);:x1=x1-1
760 FRAME
770 GOTO 860
780 IF y1=1 GOTO 640
790 FRAME:LOCATE y1-1,x1:PRINT CHR$(143);CHR$(128)
;:y1=y1-1
800 FRAME
810 GOTO 860
820 IF y1=40 GOTO 640
830 FRAME:LOCATE y1,x1:PRINT CHR$(128);:LOCATE y1+
1,x1:PRINT CHR$(143);:y1=y1+1
840 FRAME
850 GOTO 860
860 IF t=t+1 AND v=t1 GOTO 900
870 GOSUB 890
880 GOTO 840
890 t=(v-1)*1616-2*v+(25-1)*1616-2:LOCATE v,x:PRIN
T CHR$(143);:PLOT 2,V:DRAW 2+18,V:DRAW 2+18,V:18:D
RAW 2,V+18:DRAW 2,V+2=-2*v+v-2:PLOT 2,v:DRAW 2+22
,v:DRAW 2+22,v+22:DRAW 2,v+22:DRAW 2,v:RETURN
900 SOUND 1,478:SOUND 1,358:GOSUB 890:SPEED INK 10
;:OINK 1,6,26
910 IF P1=0 GOTO 930
920 PEN 2:LOCATE 1,24:q1=TIME:PRINT "CIAS -":PRI
NT USING "##.##";(q1-q)/300;:PRINT " sek"
930 a$=INKEY$
940 FOR Z=1 TO 1000:NEXT
950 RETURN
960 SYMBOL AFTER 170

```

```

970 SYMBOL 171,0,0,120,12,124,204,118,3
980 SYMBOL 172,24,60,102,102,102,102,3
990 SYMBOL 173,24,60,102,102,102,60,60
1000 SYMBOL 174,24,60,102,192,192,102,60,60
1010 SYMBOL 175,0,60,102,126,192,102,60,60
1020 SYMBOL 176,254,96,104,120,104,96,254,3
1030 SYMBOL 177,56,24,28,21,56,24,60,0
1040 SYMBOL 178,240,96,112,96,226,102,254,0
1050 SYMBOL 179,24,0,220,102,102,102,102,0
1060 SYMBOL 180,24,198,230,246,222,206,198,0
1070 SYMBOL 181,24,0,60,102,102,102,60,0
1080 SYMBOL 182,12,56,108,198,198,108,56,0
1090 SYMBOL 183,24,0,60,96,60,60,124,0
1100 SYMBOL 184,24,60,96,60,60,102,60,0
1110 SYMBOL 185,24,0,126,76,24,48,126,0
1120 SYMBOL 186,254,198,140,126,48,102,254,0
1130 SYMBOL 187,12,24,126,76,24,48,126,0
1140 SYMBOL 188,48,254,172,24,50,102,254,0
1150 SYMBOL 189,0,0,0,0,0,12,12,24
1160 KEY DEF 10,1,35,171,172
1170 KEY DEF 11,1,56,173,174
1180 KEY DEF 3,1,57,175,176
1190 KEY DEF 20,1,52,177,178
1200 KEY DEF 12,1,53,179,180
1210 KEY DEF 4,1,54,181,182
1220 KEY DEF 13,1,49,183,184
1230 KEY DEF 14,1,50,185,186
1240 KEY DEF 5,1,51,187,188
1250 KEY DEF 15,1,189
1260 RETURN

```

M. Skoneczny

Emulator Basic 1.1 dla CPC 464 (2)

Zgodnie z zapowiedzią z poprzedniego numeru przystąpię teraz do przedstawienia komend, o które wzbogacano został BASIC 1.0. **CURSOR** (przełącznik systemowy) (przełącznik użytkownika).

Funkcja ustawia przełączniki następująco są (1) i wyłączony (0). Kursor jest wyświetlany, jeśli oba przełączniki są włączone. Musi wystąpić przynajmniej jeden parametr. Nie można pominąć obu.

ON BREAK CONT i ON BREAK STOP

Pierwsza z tych funkcji likwiduje działanie klawisza ESC, którego wciśnięcie powoduje przerwanie programu. Działanie tego rozkazu jest znoszone przez drugą funkcję.

Mask (wyrażenia całkowite), (początkowe ustawienie punktu).

Rozkaz ustawia maskę i jest stosowany podczas kreślenia linii. Bi-narna wartość „wyrażenia całkowitego” z przedziału 0 — 255 ustawia bity w każdej przyległej grupie 8 punktów w stan włączony (1) lub wyłączony (0). Parametr (początkowe ustawienie punktu) określa, czy pierwszy punkt linii będzie wyświetlany (1), czy też nie (0). Musi wystąpić przynajmniej jeden z parametrów.

GRAPHICS PEN (nr tuszu), (tryb tła)

Rozkaz ustala kolor atramentu dla kreślonej grafiki (0—15). Tryb-y: 0—tło nieprzezroczyste, 1—tło przezroczyste. Tryb tła wpływa na tło znaków wyświetlanych przy aktywnym TAG oraz na kolor przetrw w liniach przy określonych parametrach instrukcji MASK.

GRAPHICS PAPER (nr tuszu)

Ustala kolor tła w trybie graficznym. Jest również kolorem tła przy zsczeniu okna graficznego (CLG).

FRAME

Umożliwia poprawę jakości obrazu przy animacji. Dzięki tej funkcji uzyskujemy płynny ruch na ekranie.

FILL (nr INK)

Wypełnia dowolny obszar okna graficznego kolorem wyspecyfikowanym w INK.

COPYCHRS (#nr strumienia)

Kopiuje znaki z bieżącej pozycji kursora znakowego z wyspecyfi-kowanego strumienia na pozycje instrukcji LOCATE.

CLEAR INPUT

Zeruje bufor klawiatury, usuwając z niego uprzednio wpisane zna-ki.

Dodatkowo, funkcje graficzne (DRAW, DRAWW, MOVE, MO-VER, PLOT, PLOTFR) występujące w BASIC-u 1.0 zostały rozzer-

zone o dodatkowe parametry — tryb tuszu i mają następującą postać: funkcja graficzna (wsp. X), (wsp. Y), (kolor tuszu), (tryb tuszu). Tryb tuszu określa sposób włączonego oddziaływania na siebie koloru tła (PAPER) i koloru tuszu (PEN). Parametr ten może przy-jmować następujące wartości:

- 0 — tryb normalny
- 1 — tryb XOR
- 2 — tryb AND
- 3 — tryb OR

Program demonstracyjny oraz przykład działania funkcji CLEAR INPUT ilustrują działania dodatkowych funkcji. Praca z emulatorem znacznie zwiększa możliwości tworzenia obrazów graficznych przy wykorzystaniu interpretera języka BASIC.

J. R.

```

10 REM*****
20 REN PROGRAM DEMONSTRACYJNY #
30 REM*****
40 HOME 1:INK 0,0:INK 1,94:INK 2,111:INK 3,26
50 WINDOW #1,25,40,1,25:PAPER #1,2:PEN #1,3:CLS #1:WINDOW #2,1,40,25,25:PAPER #1,1:IFEN #2,0:CLS #2
60 WINDOW #3,1,24,1,24
70 GOSUB 1690
80 LOCATE #2,10,1:PRINT #2,"DOKONAJ WYBORU OPCJI!";
90 a$=INKEY$:IF a$="" THEN 90
100 IF INKEY(69)=0 THEN 1470
110 IF INKEY(54)=0 THEN 1590
120 IF INKEY(62)=0 THEN 1250
130 IF INKEY(61)=0 THEN 990
140 IF INKEY(58)=0 THEN 370
150 IF INKEY(53)=0 THEN 490
160 IF INKEY(52)=0 THEN 180
170 GOTO 90
180 REN ***** CURSOR *****
190 CLS #1:CLS #2:CLS #3:GOSUB 1830
200 LOCATE #1,2,4:PRINT #1,"DEMONSTRACJA#";
210 LOCATE #1,2,6:PRINT #1," CURSOR#";
220 CURSOR 1:LOCATE #0,22,5
230 LOCATE #3,5,5:PRINT #3,"napis: litera #"
240 LOCATE #2,17,1:PRINT #2,"CURSOR 1";

```

```

250 LOCATE 83,22,5
260 AS=INKEY$:IF AS="" THEN 260
270 LOCATE 83,22,5:PRINT 80, " *":PRINT 83,8;
280 CURSOR OFF:FOR I=1 TO 2500:NEXT I
290 LOCATE 82,24,1:PRINT 82,"0";
300 LOCATE 83,5,5:PRINT 81,"napisz litere : "
310 LOCATE 83,22,5
320 AS=INKEY$:IF AS="" THEN 320
330 PRINT 83,8;
340 LOCATE 82,5,1:PRINT 82,"DOLNY KLAWISZ - POWROT DO MENU";
350 AS=INKEY$:IF AS="" THEN 350
360 GOTO 70
370 REM ##### ON BREAK #####
380 CLS #1:CLS #2:CLS #3:GOSUB 1830
390 LOCATE #1,2,4:PRINT #1,"DEMONSTRACJA";
400 LOCATE #1,2,6:PRINT #1," ON BREAK";
410 LOCATE #1,2,7:PRINT #1," CONT STOP";
420 FOR I=1 TO 1500:NEXT I
430 LOCATE 82,10,1:PRINT 82,"BREAK WYJACZONY"
440 GRAPHICS PEN 1,0:GRAPHICS PAPER 0
450 LOCATE 83,2,12:PRINT 83,CHR$(143):LOCATE 83,25,12:PRINT 83,CHR$(143)
460 TAG:MOVE 33,224
470 ON BREAK CONT
480 FOR z=1 TO 3
490 FOR x=33 TO 336
500 FRAME:MOVE x-1,224:PRINT " * ";
510 MOVE x,224
520 PRINT CHR$(243);
530 NEXT x
540 MOVE 336,224
550 FOR x=336 TO 32 STEP -1
560 FRAME:MOVE x+1,224:PRINT " * ";
570 MOVE x,224
580 PRINT CHR$(242);
590 NEXT x
600 NEXT z
610 GRAPHICS PEN 3:MOVE 384,200:DRAW 640,200:GRAPHICS PEN 1
620 LOCATE 82,16,1:PRINT 82,"wieloznakowy ";
630 LOCATE 81,2,15:PRINT 81,"po wyjsciu";
640 LOCATE 81,3,16:PRINT 81," BREAK";
650 LOCATE 81,2,17:PRINT 81,"wprowadz znak";
660 LOCATE 81,2,18:PRINT 81,"wciśnij ENTER";
670 ON BREAK STOP
680 GOTO 480
690 REM ##### MASK #####
700 CLS #1:CLS #2:CLS #3:GOSUB 1830
710 LOCATE #1,2,4:PRINT #1,"DEMONSTRACJA";
720 LOCATE #1,2,6:PRINT #1," MASK";
730 GRAPHICS PAPER 0:GRAPHICS PEN 3
740 y=377
750 lp=0
760 FOR I=1 TO 10
770 MASK #1
780 MOVE 3,y:DRAW 384,y
790 yy=I
800 lp=lp+1
810 IF lp>350 THEN 950
820 NEXT I
830 yy=I:MOVE 3,y:MASK #1:llllllllll:DRAW 384,y:yy=yy-1
840 lp=lp+1:IF lp>350 THEN 950
850 FOR I=1 TO 10
860 MASK #1:0000
870 MOVE 3,y:DRAW 384,y
880 yy=I
890 lp=lp+1
900 IF lp>350 THEN 950
910 NEXT I
920 yy=I:MOVE 3,y:MASK #1:llllllllll:DRAW 384,y:yy=yy-1
930 lp=lp+1:IF lp>350 THEN 950
940 GOTO 760
950 CLS #2:LOCATE 82,5,1:PRINT 82,"DOLNY KLAWISZ - POWROT DO MENU";
960 AS=INKEY$:IF AS="" THEN 960
970 MASK #1:llllllllll
980 GOTO 70
990 PEN ##### GRAPHICS PAPER & PEN #####
1000 CLS #1:CLS #2:CLS #3:GOSUB 1830
1010 LOCATE #1,2,4:PRINT #1,"DEMONSTRACJA";
1020 LOCATE #1,2,6:PRINT #1," GRAPHICS";
1030 LOCATE #1,2,7:PRINT #1," PAPER PEN";
1040 TAG:GRAPHICS PAPER 2
1050 GRAPHICS PEN 0,0
1060 FOR I=1 TO 2500:NEXT I
1070 MOVE 50,250:PRINT "Ala";
1080 GRAPHICS PEN 3,1
1090 FOR I=1 TO 2500:NEXT I
1100 MOVE 100,100:PRINT "As";
1110 GRAPHICS PEN 1,0

```

```

1120 GRAPHICS PAPER 3
1130 FOR I=1 TO 2500:NEXT I
1140 MOVE 300,200:PRINT "OLA";
1150 FOR I=1 TO 2500:NEXT I
1160 GRAPHICS PAPER 1
1170 GRAPHICS PEN 0,0
1180 MOVE 250,360:PRINT "DOM";
1190 FOR I=1 TO 2500:NEXT I:GRAPHICS PEN 2
1200 TAG:OFF:MOVE 150,150:DRAW 150,300:DRAW 250,350:DRAW 250,150:DRAW 150,150:HOV
E 220,220:FULL I
1210 LOCATE 82,5,1:PRINT 82,"DOLNY KLAWISZ - POWROT DO MENU";
1220 AS=INKEY$:IF AS="" THEN 1220
1230 GOTO 70
1240 REM ##### FRAME #####
1250 CLS #1:CLS #2:CLS #3:GOSUB 1830
1260 LOCATE #1,2,4:PRINT #1,"DEMONSTRACJA";
1270 LOCATE #1,2,6:PRINT #1," FRAME";
1280 LOCATE 82,10,1:PRINT 82,"funkcja FRAME aktywana";
1290 TAG:MOVE 200,50
1300 FOR y=50 TO 300
1310 FRAME
1320 MOVE 200,y-1
1330 PRINT " * ";
1340 MOVE 200,y
1350 PRINT CHR$(239);
1360 NEXT y
1370 TAG:OFF:CLS #3:GOSUB 1840
1380 LOCATE 82,24,1:PRINT 82,"wyjaczona";
1390 TAG:MOVE 320,0
1400 FOR y=50 TO 300
1410 MOVE 200,y-1
1420 PRINT " * ";
1430 MOVE 200,y
1440 PRINT CHR$(239);
1450 NEXT y
1460 GOTO 70
1470 REM ##### COPYCHR #####
1480 CLS #1:CLS #2:GOSUB 1840
1490 LOCATE #1,2,4:PRINT #1,"DEMONSTRACJA";
1500 LOCATE #1,2,6:PRINT #1," COPYCHR";
1510 LOCATE 83,5,5:PRINT 83,"SYZ";
1520 FOR I=1 TO 2500:NEXT I
1530 LOCATE 83,5,5:SUB=COPYCHR(83)
1540 LOCATE 83,10,10:PRINT 83,B;
1550 LOCATE 82,5,1:PRINT 82,"DOLNY KLAWISZ - POWROT DO MENU";
1560 AS=INKEY$:IF AS="" THEN 1560
1570 GOTO 70
1580 REM ##### FILL #####
1590 CLS #1:CLS #2:GOSUB 1840
1600 LOCATE #1,2,4:PRINT #1,"DEMONSTRACJA";
1610 LOCATE #1,2,6:PRINT #1," FILL";
1620 MOVE 50,50:DRAW 150,300:DRAW 250,150:DRAW 50,50:MOVE 150,200
1630 FOR I=1 TO 2500:NEXT I
1640 FILL 2: MOVE 260,150:FILL 1
1650 LOCATE 82,5,1:PRINT 82,"DOLNY KLAWISZ - POWROT DO MENU";
1660 AS=INKEY$:IF AS="" THEN 1660
1670 GOTO 70
1680 REM ##### MENU #####
1690 CLS #1:CLS #2:CLS #3:GOSUB 1840
1700 LOCATE #1,2,2:PRINT #1,"MENU MENU MENU";
1710 LOCATE #1,2,5:PRINT #1,"(A) COPYCHR";
1720 LOCATE #1,2,6:PRINT #1,"(B) FILL";
1730 LOCATE #1,2,7:PRINT #1,"(C) FRAME";
1740 LOCATE #1,2,8:PRINT #1,"(D) GRAPHICS ";
1750 LOCATE #1,2,9:PRINT #1," PAPER/PEN";
1760 LOCATE #1,2,10:PRINT #1,"(E) ON BREAK ";
1770 LOCATE #1,2,11:PRINT #1," CONT ";
1780 LOCATE #1,2,12:PRINT #1," ON BREAK ";
1790 LOCATE #1,2,13:PRINT #1," STOP ";
1800 LOCATE #1,2,14:PRINT #1,"(F) MASK ";
1810 LOCATE #1,2,15:PRINT #1,"(G) CURSOR";
1820 RETURN
1830 REM ##### RYSOWANIE GAMEX #####
1840 MOVE 0,16:DRAW 440,16,3:MOVE 384,398:DRAW 638,398,3:DRAW 638,16:DRAW 384,16
1850 DRAW 384,398:DRAW 0,398:DRAW 0,16
1850 RETURN

```

```

10 REM ##### CLEAR INPUT #####
20 CLS
30 PRINT "wprowadz znak z klawiatury"
40 FOR I=1 TO 5000
50 NEXT I
60 CLEAR INPUT

```

Ten słownik może się przydać wszystkim tym, którzy samodzielnie przedzierają się przez literaturę. Ci, którzy pierwsze trudności mają już za sobą wiedzą ile wysiłku wymaga zrozumienie obcych terminów technicznych i ich właściwe przetłumaczenie. Nasz słownik adresujemy do wszystkich tych, którzy odważyli się samodzielnie poszukiwać wiedzy. W następnych numerach „IKS-a”, zawsze na tych samych stronach, drukować będziemy kolejne hasła. Wytrwali Czytelnicy będą mogli skompletować pokąźnych rozmiarów angielsko-polski słownik informatyczny.

A

ABSOLUTE - absolutny, bezwzględny,
 ABSOLUTE ADDITION - dodawanie bezwzględne, dodawanie bez zwracania uwagi na znak algebraiczny,
 ABSOLUTE ADDRESS - adres bezwzględny, adres komputerowy,
 ABSOLUTE ADDRESSING - adresowanie absolutne, adresowanie bezwzględne,
 ABSOLUTE CODE - kod komputera, kod wewnętrzny,
 ABSOLUTE COORDINATES - współrzędne bezwzględne (odniesione do początku określonego systemu współrzędnych),
 ABSOLUTE INSTRUCTION - rozkaz pełny,
 ABSOLUTE LOADER - program ładujący absolutny,
 ABSOLUTE MAGNITUDE - wartość bezwzględna, moduł (liczby),
 ABSOLUTE NUMBER - liczba nieilanowana, liczba bezwzględna,
 ABSOLUTE PROGRAM - program o adresach rzeczywistych, ABSOLUTE PROGRAMING - programowanie w adresach rzeczywistych,
 ABSOLUTE VALUE - wartość bezwzględna, moduł,
 ABSOLUTE VALUE OF A COMPLEX NUMBER - moduł liczby zespolonej,
 ABSOLUTE VALUE OF A REAL NUMBER - wartość bezwzględna liczby rzeczywistej,
 ABSOLUTE VALUE OF A VECTOR - wartość bezwzględna wektora, miara bezwzględna wektora,
 ABSOLUTE VECTOR - wektor bezwzględny, (dot. grafiki),
 ABSORBED POWER - moc pobierana,
 ABSTRACT - abstrakt, streszczenie, wyciąg, skrót, analiza dokumentacyjna,
 ABSTRACT CARD - karta dokumentacyjna,
 ABSTRACT NUMBER - liczba nieilanowana, liczba oderwana,
 A.C. PLANT - urządzenie klimatyzacyjne, klimatyzator
 ACCELERATE - przyspieszać,
 ACCELERATION - przyspieszanie,
 ACCEPT - akceptować, przyjmować, także: pobrać,
 ACCEPT INSTRUCTIONS - przyjmować, realizować rozkazy
 ACCEPTABLE RELIABILITY LEVEL - dopuszczalny poziom niezawodności,
 ACCESS - dostęp, także: współpraca,
 ACCESS ARM - ramię dostępu (mechanizm zapisu lub odczytu danych z dysku),
 ACCESS CYCLE - cykl dostępu (do pamięci),
 ACCESS METHOD - metoda dostępu,
 ACCESS MODE - tryb dostępu (sposób adresowania zapisu w pliku),
 ACCESS SPEED - szybkość dostępu,
 ACCESS TIME - czas dostępu,
 ACCESSIBILITY - dostępność,
 ACCESSIBLE - dostępny,
 ACCESSING FILE - dostęp do pliku (zbioru),
 ACCESSORY - dodatkowy, pomocniczy,
 ACCLIMATE - aklimatyzować,
 ACCOMMODATE - ulokować, umieścić, zamieścić,
 ACCOMPLISH - osiągać, realizować, uzyskiwać, wykonywać,
 ACCOUNT - faktura, rachunek, także: konto,
 ACCOUNTING - księgowanie,
 ACCOUNTING DEBIT CARD - karta debetowa,
 ACCOUNTING PROGRAM - program księgujący,
 ACCUMULATE - gromadzić, zbierać (np. informacje), także: sumować, kumulować,
 ACCUMULATED ERROR - błąd sumacyjny,
 ACCUMULATING COUNTER - licznik dodający,
 ACCUMULATION - dodawanie, gromadzenie (np. dokumentów), akumulacja.

ACCUMULATOR - akumulator,
 ACCUMULATOR REGISTER - akumulator,
 ACCURACY - dokładność, wierność (np. przeniesienia, przepisania), precyzja,
 ACCURACY CONTROL CHARACTER - znak sprawdzenia dokładności,
 ACCURACY OF MEASUREMENT - dokładność pomiarów,
 ACCURACY OF READING - dokładność odczytu,
 ACCURACY OF RECORDING - wierność zapisu,
 ACCURATELY - wiernie,
 ACHIEVE - osiągać,
 ACKNOWLEDGE - potwierdzać,
 ACKNOWLEDGE CHARACTER - znak potwierdzający,
 ACKNOWLEDGEMENT - potwierdzenie,
 ACOUSTIC COUPLER - akustyczne urządzenie WEJ/WYJ,
 ACOUSTIC INDICATOR - wskaźnik akustyczny,
 ACOUSTIC MEMORY - pamięć akustyczna (na liniach opóźniających),
 ACOUSTIC MODEM - modem akustyczny,
 ACOUSTIC OUTPUT - wyjście foniczne, wyjście akustyczne,
 ACQUAINT - zapoznawać się (z czymś),
 ACQUIRE - nabywać (np. wiedzę),
 ACROSS - w poprzek (np. tadej),
 ACT - działać, spełniać funkcje, pełnić rolę,
 ACTION - działanie, także: operacja, czynność,
 ACTIVATE - uruchamiać (np. system, urządzenie),
 ACTIVATE THE DEVICE - uruchamiać urządzenie,
 ACTIVE CURRENT - prąd czynny,
 ACTIVE PAGE - stronica dostępna (część bufora ekranu dostępna dla instrukcji zapisu),
 ACTIVE-TIME - czas (data) ważności pliku (zbioru),
 ACTIVITY - czynność (np. surowania, sortowania), także: działalność,
 ACTUAL - rzeczywisty, rzeczywisty,
 ACTUAL PARAMETER - parametr aktualny,
 ACTUALLY - faktycznie, rzeczywistnie,
 ACTUATE - uruchamiać, także: pobudzać,
 ACTUATING ERROR - uchyb na wejściu czionu znacznika czcionki,
 ACTUATING SIGNAL - sygnał uruchamiający,
 ACU - patrz: AUTOMATIC CALLING UNIT,
 A-D CONVERTER - patrz: ANALOG-TO-DIGITAL CONVERTER,
 A/D CONVERSION - przetwarzanie analogowo-cyfrowe,
 ADAPT - dostosowywać,
 ADAPTATION - adaptacja, dostosowanie,
 ADAPTER - urządzenie dopasowujące,
 ADAPTING - dostosowywanie się, przystosowywanie się,
 ADAPTIVE CONTROL - adaptacyjny układ sterowania,
 ADD - dodawać, sumować,
 ADD CARRY - przeniesienie przy dodawaniu,
 ADD INSTRUCTION - rozkaz dodania,
 ADD-ON DEVICE - urządzenie dodatkowe, także: element dyskretny montowany do hybrydowego ukl. scalonego
 ADD-ON MEMORY - pamięć kumulowana,
 ADD-SUBTRACT TIME - czas operacji dodawania lub odejmowania,
 ADD UP - dodawać (liczby),
 ADDEND - dodatek, składnik dodawania,
 ADDER - sumator, układ sumujący,
 ADDER-ACCUMULATOR - akumulator przechowujący wynik dodawania,
 ADDER-SUBTRACTOR - układ sumująco-odejmujący,
 ADDING CIRCUIT - obwód sumujący,
 ADDITION - dodawanie, dodatek, uzupełnienie,
 ADDITION WITHOUT CARRY - dodawanie bez przeniesienia,
 ADDITIONAL - dodatkowy,
 ADDITIONAL CHARACTER - znak specjalny, tzn. nie litera i nie cyfra,
 ADDITIONAL INSTRUCTION - rozkaz dodatkowy,
 ADDRESSABLE MATRIX - macierz adresowalna (określa wielkość obrazu),
 ADDRESSING - adresowanie,
 ADDRESSING CIRCUIT - układ wybierania adresu,
 ADDRESSING SYSTEM - system adresowania,
 ADDRESS - adres, pozycja w pamięci,
 ADDRESS ADJUSTMENT - przesunięcie adresu, korekta adresu,
 ADDRESS ARITHMETIC - arytmetyka adresów,
 ADDRESS ASSIGNMENT - przydzielanie adresu,
 ADDRESS BUS - szyna adresowa,
 ADDRESS COMPUTATION - obliczanie adresu,
 ADDRESS CONSTANT - stała adresowa, adres podstawowy, adres początkowy,
 ADDRESS CONVERSION - konwersja adresu, przekształcanie adresu,
 ADDRESS COUNTER - rejestr (licznik) adresu,
 ADDRESS DECODER - dekodler adresu,
 ADDRESS INDICATOR - wskaźnik adresu,
 ADDRESS LATCH ENABLE (ALE) - znacznik adresu (odróżniający adres od danych),
 ADDRESS MODIFICATION - modyfikacja adresu,
 ADDRESS NUMBERING SYSTEM - system obliczania adresów
 ADDRESS PART - część adresowa (rozkażu),
 ADDRESS REGISTER - rejestr adresu,
 ADDRESS SELECTION OPERATION - operacja selekcji adresu,

ADDRESS SORTING - sortowanie adresów,
ADDRESS SUBSTITUTION - podstawienie adresu, prze-
adresowanie,
ADDRESS SYSTEM - system adresowy,
ADDRESS TABLE - lista adresów, tabela adresów,
ADDRESS TRACK - ścieżka adresowa,
ADDRESS TRANSLATION - przekształcenie adresu, kon-
wersja adresu,
ADDRESSABILITY - adresowalność, obszar adresowy,
ADDRESSABLE - adresowalny, dający się adresować,
ADDRESSABLE POINT - punkt adresowalny,
ADDRESSING - adresowanie,
ADDRESSING CIRCUIT - układ wybierania adresu,
ADDRESSING METHOD - metoda adresowania,
ADDRESSING MODE - tryb adresowania,
ADDRESSING SYSTEM - system adresowania,
ADDRESSING TECHNIQUE - metoda adresowania,
ADDRESSLESS INSTRUCTION - rozkaz bezadresowy,
ADEQUATE ACCURACY - wymagana dokładność, wierność,
ADEQUATE - adekwatny, odpowiedni, stosowny,
wystarczający,
ADEQUATE TOOL - odpowiednie narzędzie,
ADJACENT - przyległy,
ADJACENT PITCH ERROR - błąd podziakki pojedynczej,
ADJOINT - sprzężony,
ADJUTATE - sprzężony,
ADJUST - dobierać, korygować, regulować,
ADJUSTMENT - poprawianie, poprawka, porządkowanie,
regulowanie,
ADLC - patrz: ADVANCED DATA LINK CONTROLLER,
ADMISIBLE - dopuszczalny,
ADMISIBLE DECISION - decyzja dopuszczalna,
ADMISIBLE ERROR - błąd dopuszczalny,
ADMISIBLE NUMBER - liczba dopuszczalna,
ADPTION - przyjęcie (np. metody),
ADP - patrz: AUTOMATIC DATA PROCESSING
ADDRESS CONSTANT - stała podstawowa, adres początkowy
ADVANCE - postęp, także: przesuwanie się papieru,
karty (w maszynie),
ADVANCED - rozwinięty, udoskonalony, zaawansowany
(np. system),
ADVANCING - przesuwający,
ADVENTURE GAME - gra przygodowa,
ADVICE - zawiadomienie, rada, także: wiadomość,
informacja,
ADVANCED DATA LINK CONTROLLER (ADLC) - rozwinięty
sterownik łącza danych,
ADVANCED FEATURE - także: wersja rozszerzona,
ADK-LOCK PROTECTION - zabezpieczenie obrazu koń-
cowego (dot. zabezpieczania danych),
AFTERGLOW - poświata (ekranu luminescencyjnego),
AGEING - starzenie się (np. maszyny),
AGENT'S SET - stanowisko śledzenia,
AGGREGATE - agregować, gromadzić, zbierać, także:
globalny, łączny, statystyczny,
AGGREGATE CHART - zestawienie, tabela (statystyczna)
AIKEN CODE - kod AIKENA,
AIMING CIRCLE - plamka kierunkowa (służy do dokład-
nego ustawienia pióra świetlnego),
AIMING FIELD - plamka kierunkowa (służy do dokład-
nego ustawienia pióra świetlnego),
AIMING SYMBOL - plamka kierunkowa (służy do dokład-
nego ustawienia pióra świetlnego),
AIR CONDITIONED - klimatyzacja,
AIR CONDITIONING - klimatyzacja,
AL - patrz: AUTOLOAD,
ALARM SIGNAL DEVICE - urządzenie alarmowe w kompu-
terze,
ALCOR-CODE - kod ALCOR,
ALE - patrz: ADDRESS LATCH ENABLE,
ALGEBRAIC ADDER - sumator algebraiczny,
ALGEBRAICAL ADDITION - dodawanie algebraiczne,
ALGORITHM - algorytm,
ALGORITHMIZATION - algorytmizacja,
ALIAS - nazwa umowna, pseudonim,
ALIASING - użyczenie,
ALION - wyrównywać (np. papier w maszynie), także:
wyrównywać w pionie w stosunku do przeczki,
ALIGNMENT - wyrównanie,
ALIGNMENT OF RADIX POINTS - wyrównanie przecinków
zgodnie z podstawą liczenia,
ALL-INSULATED - łącznik całkowicie izolowany,
ALL POINTS ADDRESSABLE MODE (CAP) - tryb adresacji
z dokładnością do punktu,
ALL-OR-NOTHING RELAY - przełącznik dwupołożeniowy,
ALL-OR-PART SWITCH - przełącznik dwupołożeniowy,
ALLOCATE - przydzielać, przyporządkowywać, rezer-
wować,
ALLOCATION - przydział, przyporządkowanie, rezer-
wowanie,
ALLOCATION COUNTER - wskaźnik adresu,
ALLOT - przydzielać,
ALLOWABLE - dopuszczalny,
ALLOWANCE - przydział, także: przydział dodatkowy,
ALLOY JUNCTION - złącze stopowe,
ALPHABET - alfabet,

ALPHABETIC - alfabetyczny,
ALPHABETIC SYMBOL - symbol literowy,
ALPHANUMERIC - alfanumeryczny,
ALPHANUMERIC CARD PUNCH - dziurkarka kart alfanu-
mericznych,
ALPHANUMERIC CODE - kod alfanumeryczny,
ALPHANUMERIC DISPLAY - alfaskop,
ALPHANUMERIC LITERAL - literal alfanumeryczny,
ALTER - zmieniać, modyfikować (np. program),
ALTERING - dokonywanie zmian,
ALTERNATE - zmieniać,
ALTERNATE TRACK - ścieżka zapasowa (zapisu w pamię-
ci magnetycznej),
ALTERNATING CURRENT - prąd przemienny,
ALTERNATING OPERATION - transmisja naprzemienna,
transmisja półdupleksowa,
ALTERNATION - suma logiczna, dysjunkcja,
ALTERNATION OF THE LIST OF INSTRUCTIONS - zmiana cią-
gu rozkazów,
ALTERNATIVE - alternatywa, możliwość, wariant,
ALU - patrz: ARITHMETIC-LOGIC UNIT,
AMBIGUITY - dwuznaczność, możliwość wyboru dwóch
rozwiązań,
AMBIGUOUS FILE NAME - niejednoznaczna nazwa pliku
(zbioru),
AMEND - poprawiać, wnosić poprawkę, poprawka,
AMENDMENT - poprawa, naprawa (błędu),
AMPERSAND - znak "AND" (handlowe I), znak &,
AMPLIFICATION - zwiększenie, wzmocnienie,
AMPLIFICATION FACTOR - współczynnik wzmocnienia,
AMPLIFIER - wzmacniacz, wzmacniak,
AMPLIFIER ELEMENT - człon wzmacniający,
AMPLIFIER GAIN - wzmocnienie wzmacniacza, współ-
czynnik wzmocnienia,
AMPLIFIER STAGE - stopień wzmacniacza,
AMPLIFY - zwiększać, wzmacniać,
AMPLIFYING WINDING - uzwojenie wzmacniające,
AMPLITUDE ADDER - sumator amplitudowy,
AMPLITUDE OF A COMPLEX NUMBER - argument liczby zes-
polonej,
ANSTRAD DISC OPERATING SYSTEM - Dyskowy System
Operacyjny firmy ANSTRAD,
ANALOG - analogowy,
ANALOG ADDER - sumator analogowy,
ANALOG COMPUTER - komputer analogowy,
ANALOG DATA - dane analogowe,
ANALOG-DIGITAL COMPUTER - komputer hybrydowy,
ANALOG-DIGITAL CONVERSION - konwersja analogowo-
cyfrowa,
ANALOG PLOTTER - ploter analogowy,
ANALOG SIMULATION - modelowanie analogowe,
ANALOG TO DIGITAL CONVERSION - przetwarzanie analo-
gowo-cyfrowe,
ANALOG-TO-DIGITAL - przetwarzanie sygnału analogowe-
go na cyfrowy,
ANALOG-TO-DIGITAL CONVERTER - konwerter analogowo-
cyfrowy, przetwornik analogowo-cyfrowy, konwerter
A/D,
ANALOGUE - analogowy,
ANALYSING FILE ROUTINES - program drukowania rapor-
tu o zawartości pliku (zbioru),
AND - operacja logiczna "I", operator boolowski,
AND CIRCUIT - układ "I", bramka "I", układ iloczynny
logicznego,
AND-ELEMENT - element "I",
AND FUNCTION - funkcja "I", funkcja iloczynny logi-
cznego,
AND GATE - bramka "I", układ "I", układ iloczynny
logicznego,
AND OPERATION - iloczynny logiczny, koniunkcja,
ANIMATION - animacja,
ANNOTATION - uwaga, notatka,
ANSWERING - zgłoszenie,
ANTECEDENT - poprzednik,
ANTECEDENT DRIVEN - także: wnioskowanie przedzające
(w systemach ekspertowych),
ANTI-INTERFERENCE DEVICE - urządzenie przeciwdziało-
nicowe,
ANTICIPATING CONTROL - sterowanie wyprzedzające,
ANTI-VALENCE - różnica symetryczna, "LUB" wyklu-
czające,
APA - patrz: ALL POINTS ADDRESSABLE MODE,
APERTURE CARD - karta okiennowa, karta aperturowa,
mikrokarta,
API - patrz: AUTOMATIC PROGRAM INTERRUPT,
APL (A PROGRAMMING LANGUAGE) - skrót nazwy języka
programowania APL do zastosowań naukowych,
APOSTROPHE - apostrof,
APPLICATOR - aparat, przyrząd, urządzenie,
APPARENT ERROR - błąd pozorny,
APPARENT VARIABLE - zmienna pozorna,
APPENDIX - aneks, dodatek,
APPLIANCE - przyrząd, urządzenie,

Czy będziemy mieli polski mikrokomputer?

O historii polskich mikrokomputerów można by napisać poważną rozprawę. Będzie może wkrótce rozsławiał się dyskusja w poszukiwaniu odpowiedzi dlaczego nam się nie udało. Argumentów z pewnością nie zabraknie.

Po chworskim Juniorze, po reklamowanym w Krakowie 86 i okrzykanej Mazovi pojawił się na wybrzeżu BOSMAN-8. Mikrokomputer skonstruowano w ubiegłym roku; pierwsze prototypy pokazano na Targach Poznańskich zapowiadając możliwość rychłego uruchomienia seryjnej produkcji w gdańskim UNIMORze.

Produkcji, ale czego? BOSMAN-8 jest mikrokomputerem osmiobitowym zbudowanym w oparciu o procesor Z80 A. Posiada pamięć RAM o pojemności 512 KB i 16-kilobytów EPROM. Standardowo ma dwa napędy dysków elastycznych 3, 3,5 lub 5 1/4 calowych. Pracuje pod nadzorem systemu operacyjnego CPM/R (zgodny z CPM 2.2) rezydującym w ROM-ie, który umożliwia dołączenie dowolnych urządzeń zewnętrznych z interfejsem szeregowym V24 lub równoległym (Centronics, DZM-180).

Istotną zaletą tego komputera jest dysk krzemowy o pojemności 416 KB, który znacznie usprawnia pracę programów często odwołujących się do pamięci dyskowej. Zwiększenie szybkości i niezawodności mikrokomputera uzyskano również przez umieszczenie systemu operacyjnego w pamięci EPROM.



To oczywiście nie wszystkie niespodzianki. W pamięci stałej rezyduje również program poprawiający sprawność komunikacji użytkownika z komputerem. Przed wszystkim pozwala na przetwarzanie pliku komend zwanego plikiem wsadowym, umożliwia wprowadzenie informacji na drukarkę równoległą z wykonywaniem innego programu czyli tzw. drukowanie w tle.

Program D łączy w sobie możliwości systemowych poleceń CP/M takich jak: DIR, ERA, REN, PIP, STAT, TYPE, DUMP, USER.

BOSMAN-8 jest łatwy w obsłudze, dobre oprogramowanie sprawia, że nadaje się on szczególnie do przetwarzania danych w biurach i małych przedsiębiorstwach, przetwarzania tekstów, obliczeń inżynierskich, prac projektowych i programowych (kompilatory języków: Fortran, Basic, Cobol, C. Forth, Pascal), organizacji sieci lokalnych.

Jeszcze jeden ważny szczegół, BOSMAN-8 z odpowiednim oprogramowaniem może pracować jako końcówka du-

żych komputerów z wielodostępem np. R-32, IBM-360, VAX, ODRA.

Inaczej mówiąc konstruktorzy włożyli wiele wysiłku, aby z początku już przecież Z80 A wyśnawić wszystko co można. I tu kolejny już raz w historii polskiej myśli technicznej należy schylić czoła przed pomysłowością i odwagą naszego konstruktora. BOSMAN-8, pracownikiem Instytutu Okrętowego Politechniki Gdańskiej (bo oni właśnie są konstruktorami tego mikrokomputera), udał się.

Tu jednak wyraźnie podkreślić trzeba, że BOSMAN-8 IBM-em nie jest! Będzie natomiast sukcesem, jeśli uda się uruchomić jego masową produkcję. Konstruktorzy mają swoje racje. Sprzęt klasy IBM nie jest przecież wszystkim potrzebny, a jego obecność w wielu miejscach byłaby po prostu marnotrawstwem. Nasza rodzima informatyka opierać się będzie w najbliższej i chyba, niestety, dalszej przyszłości o lokalne systemy informatyczne. Na budowę rozbudowanych sieci przy obecnym stanie łączności nas po prostu nie stać. Na przykład w wielu przedsiębiorstwach można by sieć konstruować w oparciu właśnie o BOSMAN-a, pozostawiając komputer lepszej klasy jako jednostkę centralną. Okazuje się, że BOSMAN-8 całkiem dobrze pracuje jako terminal szesnastobitowego IBM-a. Ale to są problemy techniczne. W wielu przypadkach drugorzędna sprawą jest czas uzyskania informacji. Zysk rzędu kilku nawet minut jest w przypadku naszej organizacji niezauważalny. Na przykład zestawienia statystyczne przygotowane na nowoczesnym DRS300 w czasie paru minut (firma ICL) mogą zupełnie spokojnie przeleżeć na biurku referenta nawet kilka dni, zupełnie go nie drażniąc.

Czy dzięki BOSMANOWI zmieni się informatyczny krajobraz naszego kraju?

inż. Andrzej NOWAK

z-ca dyr. Instytutu Okrętowego Politechniki Gdańskiej

— BOSMAN powstał z konieczności. Okazało się, że zrobienie własnego komputera jest tańsze niż zakupienie takiego urządzenia np.

Gdańskie Zakłady Elektroniczne UNIMOR są wyspecjalizowanym przedsiębiorstwem polskiego przemysłu elektronicznego.

Zatrudniają liczną kadrę fachowców dla potrzeb wielokopijnej produkcji telewizyjnej oraz profesjonalnych urządzeń radiokomunikacji morskiej i lotniczej.

Rozpoczęto również przygotowania do produkcji sprzętu informatycznego, który obok znanych i nieposzechnych monitorów „Neptun”, stanowić będzie oferty produkcyjną UNIMOR.

Nawet odbiorniki telewizyjne, zarówno monochromatyczne, jak i kolorowe, są stale doskonalone i dostosowywane do wymagań klientów.

Oprócz nowoczesnych układów elektronicznych, odbiorniki te posiadają estetyczne kształty, dostosowane do gustów odbiorców i mody panującej na rynku. Mogą być wyposażone w zdalne sterowanie (na podczerwieć) oraz odbierać programy w systemach PAL i SECAM.

Odbiorniki turystyczne mogą być zasilane z akumulatora samochodowego, posiadają również własną antenę.

Produkcja malosygnalna to przede wszystkim radiostacje ratunkowe dla statków, wyposażenie radiokabiny okrętowej, radiostacje kutwów itp.

Dla samolotów sanitarnych i lotniczych oraz szybowców produkujemy małe radiostacje typu RS.

w Zabru. Sądzę, że udało się zrobić mikrokomputer nowoczesny i niezawodny. Nasze prototypy nie uprzykrzyły jeszcze twórców w zakładowaniu.

Oczywiście nie jest to światowa nowość. Nasza propozycja jest optymalnym rozwiązaniem na miarę możliwości materialnych i technologicznych, a także rzeczywistych potrzeb.



inż. Mieczysław BARGIEŁ
kierownik Zakładowego Ośrodka Opracowań i Produkcji Doświadczalnego Sprzętu Informatycznego (UNIMOR)

— Od prototypu do seryjnej produkcji droga jeszcze daleka. UNIMOR jest w stanie BOSMAN-a robić. Oczywiście istnieje jeszcze problem kosztów. W tej chwili podstawowa konfiguracja tego mikrokomputera kosztowałaby w granicach 900 — 1300 tys. Przy uruchomieniu produkcji seryjnej być może daloby się ją obniżyć. Z pewnością taniej kosztowałyby nas dolarowe podzespoły (choćbyśmy napędzi). Dotychczas kupujemy je pojedynczo, a taki zakup jest z reguły droższy od hurtu. Poza tym nie zawsze możemy liczyć na rodzinnych kooperantów, nie wszyscy potrafią uruchomić nowoczesną produkcję.

Pocztowa giełda

Rozwiązanie krzyżówki z nr. 4/87 „IKS-a”.
Hasło brzmi: ELWAT, ZAM, ODDA, KRAK — komputery

Bony pieniężne (1000 zł) wylosowali: Krzysztof Zieliński — Łódź, Zdzisław Ziolkowski — Łódź, Sławomir Sobczak — Częstochowa, Marian Śmigiełski — Kęty, Adam Grodzki — Warszawa.

Nagrody książkowe otrzymują: Ryszard Leś — Gdańsk, Wojciech Wierzbicki — Katowice, Stanisław Bloński — Hłyżne, Joanna Pasternak — Brzeg, Leszek Pilańczyński — Łaski Wielkie.

Rozwiązanie krzyżówki z nr. 5/87 „IKS-a”

Hasło brzmi: „IKS” propagatorem wiedzy i techniki mikrokomputerowej
Bony pieniężne (1000 zł) wylosowali: Barbara Karłowska — Wrocław, Grzegorz Kowalski — Sandomierz, Robert Grochala — Rzeszów, Maria Gajewska — Szczecin — Piotr Jakubiec — Hrubieszów.

Nagrody książkowe otrzymują: Maria Koniczna — Leszno, Leszek Jerzewski — Łódź, Marek Olesz — Gdynia, Patrycja Blaszk — Częstochowa, Sławomir Bukowski — Pruszków, Michał Pańniewski — Stawiguda woj. olsztyńskiego, Iwona Kostro — Białe Szczepanowice, Tomasz Tęcza — Jelenia Góra, Anna Wróbel — Sieradz, Elżbieta Pawelec — Przasnysz.

Rozwiązanie krzyżówki z nr. 6/87 „IKS-a”.
Hasło brzmi: W podróży z „IKS-em”
Bony pieniężne (1000 zł) wylosowali: Małgorzata Zylńska — Szczecin, Paweł Malarski — Warszawa, Piotr Hoppel — Poznań, Karol Klepacz — Łódź, Grażyna Kijowska — Szczecin.

Nagrody książkowe otrzymują: Marek Szymocha — Poręba, Wiesława Laskowska — Poznań, Elżbieta Marlewska — Gdańsk, Ewa Tiahybok — Kraków — N.Huta, Tomasz Smoleński — Szczecin, Andrzej Ryzkiewicz — Rawa Mazowiecka, Ireneusz Grzybek — Walbrzych, Krzysztof Gajda — Kraków, Mariusz Gruszczyński — Dębica, Tomasz Sobczak — Żyrardów.

Sharp MZ-800: wymienie programy, literaturę. Zbigniew Zajac, ul. 1 Maja 13 37-500 Jarosław.

Atari, Commodore, Spectrum — sprzedaż, wymiana literatury, także po polsku. Informacje — znaczki 30 zł. Klub MIKROPOL, 86-300 Grudziądz 1, skrytka 49.

Kupię układ scalony ROM 8 kB, 2364 CT 2, 27-0152-03-01 rod produkcji 1983. Piotr Elwart, 80-287 Gdańsk-Wrzeszcz, ul. Migowska 18/2.

Poszukuję zeszytów „Wszystko o komputerze” nr 3, 4, 5, 6. Artur Zabecki 33-300 Nowy Sącz, ul. Klaszorna 28/99.
Wymienie programy na Commodore C-64. Bartosz Muszyński 75-519 Koszalin, ul. Hanki Sawickiej 166/3, tel. 521-11.

Mikrokomputer Atari 800 XL — ponad 100 programów, w tym bardzo ciekawe — wymienie lub tanio odstąpię. Radostaw Wójcicki 55-035 Oborniki Sl. ul. F. Dzierżyńskiego 3/2.

LIGA MYŚLĄCYCH

Zadanie 1

Nazwijmy liczbą symetryczną taką liczbę naturalną, której cyfry stojące na miejscach pierwszym i ostatnim; drugim i przedostatnim itd. są takie same np. 3553, 474. Ile jest liczb symetrycznych trzycyfrowych, a ile liczb symetrycznych czterocyfrowych?

Zadanie 2

Najwdzię trzy kolejne liczby naturalne o tej własności, że jeżeli kwadrat największej z nich podzielimy przez liczbę najmniejszą i od otrzymanego ilorazu odejmiemy 6, to otrzymamy liczbę większą od pozostałej z szukanych liczb?

Zadanie 3

W jakim układzie liczenia prawdziwa jest równość:

$$300 - 233 = 1$$

Zadanie 4

W jednej ze szkół podstawowych odbyły się zawody sportowe. W rywalizacji indywidualnej każdy uczestnik przeprowadził z każdym z pozostałych po 3 próby sprawności. Wszystkich prób było 135. Ile uczniów brało udział w zawodach?

Zadanie 5

Grupa członków Warszawskiego Towarzystwa Cyklistów wybrała się na wielodniową wycieczkę rowerową. Zaplanowano przebieg na rowerach trasy 360 km, jadąc codziennie taką samą liczbę kilometrów. Faktycznie jednak grupa przedzieliła każdego dnia o 4 km mniej i z tego powodu wycieczka przedłużyła się o 3 dni. Ile dni miała trwać wycieczka?

Rozwiązania zadań prosimy przesyłać do redakcji do końca listopada br. z dopiskiem „Liga Myślących”. Punktacja zależy od liczby prawidłowych rozwiązań. Wśród uczestników rozlosujemy książki, a na zwycięzców „Ligi” czekają dodatkowe cenne nagrody-niespodzianki.

ZX — GEODEZJA

```
100 ZENKOS PAPER 7: BRIGHT 4: s=2
101 ZENKOS V ? : ; w:
102 INPUT PAPER 7: BRIGHT 1: M
103 LYSKOSDC V ? : ; w: LET skw=17
104 /w:
105 IF skw<=sks THEN LET sskala=s
106 IF skw>=sks THEN LET sskala=s
107 D IN X(50): D IN Y(50)
108 FOR n=1 TO 50: BRIGHT 1: X
109 INT (252/skala): " X(n)
200 IF X(n)/skala>255 THEN BEEP
1-6: GO TO 130
210 PRINT OVER 1: TAB 4: "X": n:
1-6: OVER 1: X(n):
220 INPUT PAPER 7: BRIGHT 1: Y
1-6: INT (172/skala): Y(n)
230 IF Y(n)/skala>175 THEN BEEP
1-6: GO TO 220
240 PRINT OVER 1: TAB 20: "Y": n:
1-6: OVER 1: Y(n):
250 IF n=1 THEN PLOT 1+skala*(X
n)-Y(n)+w(n)
260 IF n=1 THEN DRAW -skala*(X
n)-Y(n)+w(n)
270 IF X(n)*X(n)+Y(n)*Y(n)+w(n)
AND n>1 THEN GO TO 290
280 NEXT n
290 LET d=0
310 FOR i=1 TO n-1
320 LET f=(X(i)-X(i+1))*Y(i)
```

```
+Y(i+1))/2
330 LET d=d+SOR ((ABS (X(i)-X(i
+1))+ABS (Y(i)-Y(i+1))+2)
340 NEXT i
350 LET f=ABS f: PRINT OVER 1: I
NOS BEEP :2: f: THEN COPY
370 PRINT OVER 1: INK 3: Obw
od wznosi " f:
380 FOR f=10 TO 170 STEP 10: PL
OT 0: f: DRAW 2:0: NEXT f
390 INPUT PAPER 7: BRIGHT 1: D
RUKORC /? N ? : : LINE d$: IF d$=
" OR d$=" THEN COPY
400 PRINT OVER 1: FLASH 1: B
RIGHT 1: NACISNIJ DODOLNY KLAWIS
Z" : PAUSE 3: CLS : GO TO 0100
410 REM "Geodezja"
413 CLS : BEEP :2:14: BEEP :2:2
: BORDER 6: PLOT 26:20: DRAW 0:0
0: DRAW 130:-50: DRAW -130:0
420 PAUSE 10: PLOT 240:155: DR
AW -50:0: DRAW 80:-155: DRAW 0:15
5
430 PAUSE 10: PLOT 60:40: DRAW
140:0: DRAW 0:75: DRAW -140:-75
440 PAUSE 10: PLOT 15:165: DRAW
80:0: DRAW -80:-50: DRAW 0:59
450 PAUSE 20: PRINT AT 7:4: INK
3: BRIGHT 1: PAUSE 1: CLS
E 2: 0: PAUSE 0: CLS
460 FOR u=0 TO 7: READ v: POKE
U+3: v: NEXT v
470 DATA 55.0,16.32,56.0,0.0
480 REM "w"
490 PLOT 230:69:25
500 GO TO 100
5080 REM "Geodezja"
5090 INPUT "Geodezja" : LINE 410:
5100 PAUSE 1: "uerifu" : Geodezja--: U
ERIFY --Geodezja--: GO TO 410
```

KUPON



KRZYŻÓWKA NR 9



POZIOMO: 1) piwo domowej roboty, 5) członek męskiego kwartetu wokalnego, 6) relacja, meldunek, 8) próżniak, walkoń, 9) ogół pracowników zatrudnionych w jednostce produkcyjnej, 11) do wbijania gwoździ, 12) zgromadzenie zakonne; zakon, 13) górzysta wyspa koło Australii,

PIONOWO: 1) ogrodzenie z desek, 2) związek chemiczny lub ich mieszanina, przygotowane do jakiegoś specjalnego użytku, 3) wielokat, 4) nakład pieniężny na określony cel, 6) fizyczne zjawisko związane z ruchem drgającym lub falowaniem; wespółdranie, 7) mała półka, 10) uszkodzenie, zepsucie się maszyny, urządzenia technicznego, 11) sportowa gra mieszana w tenisie.

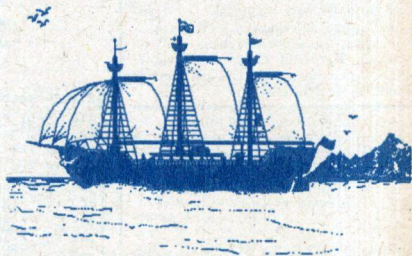
Rozwiązania krzyżówki na kartach pocztowych należy nadsyłać pod adresem redakcji do końca listopada 1987 r., naklejając kupon „IKS-a”. Wśród czytelników rozlosuje się bony pieniężne i nagrody książkowe.

MIKROCIEKAWOSTKI



Kolejna wersja AMSTRADA

Amstrada PC 1640 ECD zbudowano na mikroprocesorze 8086. W porównaniu z poprzednikami ma znacznie zwiększone możliwości graficzne: dwa tryby rozdzielczości 320 na 200 lub 640 na 200 linii w 4 kolorach zapewniają dobrą jakość prac graficznych. Systemem operacyjnym PC 1640 jest MS DOS 3.2.



Rysunek z kolekcji GEM-u wykonany na zestawie firmy ICL:

komputer — DRS 300
drukarka mozaikowa ML 192

„IKS” — dodatek „Żołnierza Wolności”. Redagują: Wiesław Cetera (kierownik zespołu), Ryszard Rogoń. Stali współpracownicy: Włodzimierz Gogolek, Janusz Janiec, Krzysztof Mamcarz, Ireneusz Miernik, Janusz Miller, Jacek Szaniawski. Adres redakcji: 00-950 Warszawa ul. Grzybowska 77, telefon centrali 20-12-61 w. 486. Telex 313664. Rękopisów nie zamówionych redakcja nie zwraca i zastrzega sobie prawo do skrótów. Nakładem Wydawnictwa „Czasopisma Wojskowe”, Warszawa ul. Grzybowska 77, Fotokład i druk rotograwiurowy — Wojskowe Zakłady Graficzne im. gen. dyw. A. Zawadzkiego. Nr zam. 9296. Nr ind. 361682.

Edytor graczy-pocisk

Tomasz MROWIEC
Ludwik PIEŁA

Mozemy wyrzucić papier milimetryowy i tablicę konwersji dwójkowo-dziesiętnej. Program nr 4 umożliwi projektowanie graczy i pocisków na ekranie za pomocą manipulatora. Dzięki jego odpowiedniej konstrukcji, łatwo możemy przedstawić graczy dowolnej wielkości, włączając obiekty o podwójnej szerokości lub łączenie dwóch graczy w jednego. Po wykonaniu tego możemy obejrzeć gracza „w akcji”, a nawet automatycznie wygenerować procedurę w języku BASIC do używania graczy i pocisków we własnych programach. Wszystkie możliwości programu opisane są w dalszej części.

Używanie edytora

Obejrzymy prosty przykład użycia programu. Wybierzmy gracza o rozdzielczości dwuwierszowej, pojedynczej szerokości i długości ośmiu linii. Umieścimy kursor w prawym, górnym rogu sztalugi, naciśniemy „K” dla wybrania kreślenia, przesuniemy kursor do lewego, dolnego rogu i naciśniemy klawisz. Mamy gracza składającego się z przekątnej. Naciśniemy „S” w celu zatrzymania, po wyświetleniu menu naciśniemy „L”. Komputer wyświetli teraz gracza w postaci danych numerycznych. Zauważmy, że każda liczba reprezentuje wartość bitową punktów elementarnych G-P w miarę przesuwania się w dół gracza, linia po linii. Aby zobaczyć naszego gracza w trybie grafiki G-P, należy nacisnąć „0”, po kilku sekundach pojawi się on na ekranie.

A co będzie, gdy zechcemy zobaczyć nasz obiekt w rozdzielczości jednowierszowej i o wielkości cztery? W oknie tekstowym pojawiają się odpowiednie instrukcje i bez trudu mogą być przeprowadzone odpowiednie zmiany. Założymy, że chcemy dokonać zmiany obrazu. Po kilku naciśnięciach klawiszy znowu jesteśmy na sztaludze. Różni się ona teraz od oryginalnej, zarówno wielkością, jak i kształtem, odzwierciedlając zmiany. Program zapewnia całkowitą swobodę przy przechodzeniu z jednego obszaru do drugiego. Umożliwia to wykonanie tylu zmian, ile potrzeba, przed zapisaniem ostatnich danych obrazu na taśmie. Możemy dodać lub wykasować linie u góry lub u dołu, a procedura wypełniania umożliwi wypełnienie obszaru.

Poeksperymentujmy z łączeniem dwóch lub czterech graczy, zmianą parametrów wielkości i rozdzielczości, a uzyskamy lepsze zrozumienie znaczenia i wzajemnych relacji tych parametrów.

Jednym z unikalnych narzędzi używanych w programie jest wykorzystanie funkcji SGN i operatorów logicznych do przekształcania odczytanych wartości STICK(0) na współrzędne X, Y. Aktualne przykłady w programie mogą być trochę niejasne. Procedura jest następująca:

```
10 S=STICK(0)
20 IF S<12 THEN X=X+SGN (8-S):
: X=X+(X<XMIN)-(X>XMAX): S=S+
: (X<8):4+4
30 IF S<15 THEN Y=Y+SGN(13.5-S):
: Y=Y+(Y<YMIN)-(Y<YMAX)
```

Mozecie nie wierzyć, lecz te trzy linie odczytują dziewięć pozycji manipulatora, tłumacząc odczyty na nowe współrzędne X, Y i dopasują te współrzędne do granic wyrażonych przez wartości X i Y, maksymalne i minimalne. Dla nieobitych z instrukcjami logicznymi, wartość w nawiasach porównania lub równości staje się jedynką, jeśli warunek jest spełniony i zerem, jeśli niespełniony. Takie instrukcje mogą często zredukować dwie linie kodu, używając instrukcji IF, do jednej linii.

Zwróćmy uwagę na ostrzeżenie, jeśli planujemy wykorzystanie programu. Przeznaczony on jest do użycia z rozdzielczością jednowierszową. Adresy dla graczy, w pamięci G-P, indeksowane są przez numer strony. Jednak przy rozdzielczości dwuwierszowej pamięć dla graczy dwa i cztery rozpoczyna się w odstępnie połowy strony i nie może być osiągnięta. Program kreślenia G-P obchodzi to ograniczenie poprzez rozszerzenie długości graczy jeden i trzy w celu włączenia graczy dwa i cztery odpowiednio. Działa to jednak tylko wtedy, gdy sąsiedni gracze przesuwani są razem.

Instrukcja programu

A. Pytania początkowe

1. Wprowadzenie rozdzielczości. Rozdzielczość wszystkich graczy i po-

cisków musi być taka sama w jednym programie.

2. Wprowadzenie gracza lub pocisku. Gracz ma szerokość ośmiu punktów elementarnych G-P, pocisk ma szerokość dwóch punktów.

3. Wprowadzenie liczby graczy. Mamy do dyspozycji czterech graczy, każdy o szerokości ośmiu punktów G-P. Można podać dowolną ilość graczy umieszczonych obok siebie i przesuwanych razem.

4. Wprowadzenie wielkości. Parametr ten może być zmieniony później w programie.

5. Wprowadzenie liczby linii pionowych. Linie mogą być dodane lub wykasowane później w programie.

B. Sztaluga kreślenia

Sztaluga kreślenia ma kolor zielony. Kreślony gracz pojawi się w kolorze pomarańczowym. Kursor wskazywany jest przez jaśniejszy zabarwiony obszar zielony lub pomarańczowy. W celu zmiany koloru należy nacisnąć przycisk manipulatora. Do przesuwania kursora używamy manipulatora.

Opcje klawiatury:

(0-9) Steruje szybkością ruchu kursora. 0-szybko, 9-wolno, 2-szybkość początkową.

(K) kreślenie: kreśli linie od aktualnej pozycji kursora. Przesuwny kursor do nowej pozycji i naciśniemy przycisk w celu wykreślenia linii między dwoma punktami.

(W) wypełnianie: Używane do wypełnienia obszaru. Musi on być ograniczony przez kolor pomarańczowy na wszystkich bokach. Należy umieścić kursor poniżej najwyższej zielonej spacji i jeśli to możliwe, powyżej najniższej zielonej spacji oraz nacisnąć W. Dla obszarów o urozmaiconym kształcie procedura ta może wymagać powtórzenia.

(L) linie: Dodaje lub kasuje linie od góry lub dołu sztalugi.

(S) top: Zatrzymuje kreślenie. Po wykreśleniu skomplikowanego rysunku może być zauważalne oczekiwanie podczas tłumaczenia diagramu na dane numeryczne.

C. Menu opcji

(O) bejrzeć: Obserwacja gracza w grafice G-P. Przesuwamy gracza przy użyciu manipulatora. W czasie realizacji tej opcji można nacisnąć:

(W) wielkość — do zmiany wielkości gracza.

```

ZL 4 REM *****
FG 5 REM *
AR 6 REM * Program nr 4 *
FI 7 REM *
ZP 8 REM *****
UE 10 DIM AS(1),P(3),PY(3),PX(3):
PR=2:LA=1990:CO=24:PLX=53248:P
L1=1780:PL1=1784
DX 15 IF PEEK(1536)<=162 THEN GOS
UV 20 PM=PEEK(106)-16:PMB=#256:
POKE 54279,PM:FOR A=0 TO 3:PC(A
)=PMB#A24:NEXT A
GW 30 GRAPHICS 0:?"? "PODAR ROZD
ZIELCZOSCI PIONOWA "1? "1 " (1
)=KOMBIERSZOWA "1? "1 " (2)=JE
DNOWIERSZOWA "1? "1 "
TX 40 TRAP 30:INPUT RE:IF RE<2 OR
RE<1 THEN 30
DI 50 ? " ? "CZY CHCESZ DEFINIOWAC
GRACZA CZY:?"? "JESLI:"TRAP
50:INPUT AS:IF AS<"G" AND AS<
">:"P" THEN 50
OE 50 IF AS="P" THEN MI=1:NP=1:GO
TO 80
BK 70 ? " ? "ILU GRACZY CHCESZ DEF
INIOWAC (1-4)?:TRAP 70:INPUT
NP:IF NP<1 OR NP>4 THEN 70
IG 60 NP#NP-1:?"? "PODAR ROZMIAR
POZIOMY (1,2 LUB 4)?:TRAP 80
:INPUT SI:IF SI<1 OR SI>4 OR S
I=3 THEN 80
RK 90 GOSUB 170
DR 100 ? " ? "ILE LINII BEDZIE MIA
L GRACZ (1-24)?:TRAP 100:INPU
T VL:IF VL<1 OR VL>24 THEN 100
NA 110 GOSUB 200:SETCOLOR 2,12,6:
Y:COLOR 3:X=208:Y=108:1:PLOT X,
Y:DRAWTO X#D,Y
KB 120 IF STRIG(0)=0 THEN POKE 71
0,42+(PEEK(710)+42)*156:PLOT X,
Y:DRAWTO X#D,Y
JL 130 S=STICK(0):IF S<15 THEN GO
SUB 250
VZ 140 IF PEEK(53775)<255 THEN 30
SW 150 FOR A=1 TO 25:PA:NEXT A
MO 160 GOTO 120
UG 170 IF RE<NP+1)*SI<1 THEN RE
TURN
BP 180 GRAPHICS 0:?"? "NIE MOZE
ZDEFINIOWAC 4 GRACZY O ROZDZIE
LZCOSCI JEDNOWIERSZOWEJ I ROZM
IARZE >?:
TP 190 ? " ? "ROZMIAR BEDZIE ZMIENIO
NY NA 2, "SI=2:?"? "NARCISNIJ
RETURN:INPUT AS:RETURN
BP 200 XD1=#SI:XD=XD1-1:H=#NP+1
:XD1:IF NI THEN H#H+4
AI 210 G=1+(CHD5) OR (VL>20)+2*(
H<10) OR (VL<40):GRAPHICS 19
<G>1#G
UI 220 YMIN=INT(108*VL/2):YMAX=I
NT(108*VL/2-1):XMIN=208*4#H:
YMAX=208*4#H-1:XD
VA 230 SETCOLOR 0,2,8:SETCOLOR 1,
12,4:COLOR 2:FOR Y=YMIN TO YMA
X:PLOT X#D,Y:DRAWTO X#A#X#D,Y
NEXT Y:RETURN
VM 250 GOSUB 290
RO 260 IF S<12 THEN X#X+SGN(8-S)*
XD1:Y#Y+(X#GMIN)*XD1-(X#YMAX)*
XD1:Y#Y+(S<8)*Y+4
ZE 270 IF S<15 THEN Y#Y+SGN(13,5-
S)*Y#Y+(Y#YMIN)-(Y#YMAX)
TQ 280 LOCATE X,Y:A:POKE 710,PEEK
(787#A)+2:COLOR 3:PLOT X,Y:DRA
WTO X#D,Y:Y:RETURN
JB 290 COLOR 14<=PEEK(710)=198:PL
OT X,Y:DRAWTO X#D,Y:Y:RETURN
VW 300 GOSUB 400:IF A#ASC("K") TH
EN 450
CJ 310 IF A#ASC("M") THEN 600
QU 320 IF A#ASC("S") THEN GOSUB 7
00:GOTO 500
RN 330 IF A#ASC("L") THEN GOSUB 7
00:GOTO 750
UX 340 IF A#ASC("A" OR "D") THEN PA#A
=47
MO 350 GOTO 120
IY 400 OPEN #1,4,0:"K":#GET #1,A:
CLOSE #1

```

```

WX 410 IF PEEK(53755)<255 THEN 41
0
ZE 420 RETURN
BZ 450 X1=X:Y1=Y
ZO 460 S=STICK(0):IF S=15 THEN 46
0
UI 470 GOSUB 260
UV 480 IF STRIG(0)=0 THEN 510
KA 490 S=STICK(0):IF S<15 THEN GO
SUB 250
PY 500 GOTO 480
WS 510 COLOR 1:X2=X:Y2=Y:Y3=Y1:O
M#1SGN(X2-X1):B=SGN(Y2-Y1):C=
X2-X1#A#D#Y2-Y1+B
SM 520 IF C=0 OR B=0 THEN FOR A=0
TO XD:PLOT X1+A,Y1:DRAWTO X2+
A,Y2:NEXT A:GOTO 570
UU 530 FOR X#X1 TO X2 STEP A:Y#Y
Y#Y2 TO Y2 STEP B:PLOT X,Y:DRA
WTO X#D,Y
HB 540 IF (X-X1+A)/C=(Y-Y1+B)/D T
HEN Y#Y+B:GOTO 560
RS 550 IF (X-X1+A)/C=(Y-Y1+B)/D T
HEN NEXT Y
BF 560 Y#Y+NEXT X:X#X+A:Y#Y+B
DU 600 GOSUB 290:GOTO 120
SH 610 X1#X:COLOR 2:PLOT X,Y:DRAW
TO X#D,Y:Y#Y+670
RS 610 GOSUB B:LOCATE X,Y,1:A:IF
A#2 THEN Y#Y-1:GOTO 610
TQ 620 GOSUB B:LOCATE X-1,Y,A:IF
A#2 THEN X#X-1:GOTO 620
IB 630 GOSUB B:COLOR 1:PLOT X,Y
US 640 GOSUB B:LOCATE X+1,Y,A:IF
A#2 THEN X#X+1:PLOT X,Y:GOTO 6
40
PB 650 X#X1:Y#Y+1:GOSUB B:LOCATE
X,Y,A:IF A#2 THEN 620
EU 660 GOSUB 290:GOTO 120
UV 670 IF X#YMAX OR X#YMIN OR Y#Y
MAX OR Y#YMIN THEN POP :GOSUB
280:GOTO 120
ZS 680 RETURN
FA 700 GOSUB 290:FOR A=0 TO NP:#B
=0:X1=XMIN+8#A#XD1:FOR Y=YMIN T
O YMIN+VL-1
DO 710 C=128#D#0:FOR X=X1 TO X1+7
:XD1 STEP XD:IF NI THEN FOR X
=X1 TO XMIN+XD1 STEP XD1
WN 720 LOCATE X,Y,E:IF E=1 THEN D
=D+C
PB 730 C=C/2:NEXT X:B=B+1:POKE P(A
#B),D:NEXT Y:NEXT A:RETURN
LV 750 D#D=I#0:?"? "PODAR LUB
KASU LINII:INPUT AS:IF A#="W
" THEN D#1:GOTO 770
LR 760 IF A#<"0" THEN 770
TP 770 ? " ? "U DORY CZY U DOLU:1:
INPUT AS:IF AS="G" THEN T=1:GOT
O 790
NN 780 IF A#<"0" THEN 770
KZ 790 ? " ? "ILE LINII:TRAP 790
:INPUT C:IF D THEN C=C-(C-VL-1
):C#C#VL:GOTO 840
BB 800 IF VL<24 THEN C=24-VL:?"
? "ZA DUZO?:?"? "MDEZSE DODA
C TYLKO "C": LINII"? :?"? "NA
CISNIJ:RETURN:INPUT AS
MC 810 IF THEN FOR A=0 TO NP:FO
B=C:BEVL TO 0 STEP 1:1:POKE P(A
+B),C:NEXT B:NEXT A
DR 850 FOR A=0 TO NP:FOR B=VL-C+1
TO VL:POKE P(A+B),0:NEXT B#C
XT A:VL=VL-C:GOTO 1400
GZ 860 FOR A=0 TO NP:FOR B=1 TO C
:POKE P(A+B),0:NEXT B:NEXT A
ET 870 RETURN
VT 870 FOR A=0 TO NP:FOR B=VL-C+1
TO VL:POKE P(A+B),0:NEXT B#C
XT A:RETURN
OV 900 GRAPHICS 0:?"? "CO CHCESZ
ZROBIC:?"? "JERZEC GRAC
ZA??"? "CHR*(29):?"? "PISAC
DRANE

```

```

CZ 905 ? CHR*(29):?"? "PISWIETLIC DR
ANE
NL 910 ? CHR*(29):?"? "OPRACNIK GRAC
ZA??"? "CHR*(29):?"? "ROZPOCZAC Z
NOMY GRACZEN?:"? "CHR*(29):?"? "S
TOP"
FT 920 GOSUB 400:IF A#ASC("0") TH
EN 1000
CN 930 IF A#ASC("2") THEN 1500
EX 940 IF A#ASC("1") THEN 1450
KO 950 IF A#ASC("P") THEN GOSUB 1
70:GOTO 1400
NN 960 IF A#ASC("R") THEN RUN
UF 970 IF A#ASC("S") THEN STOP
PG 980 GOTO 900
XD 1000 TRAP 400#0:GOSUB 1240
RS 1010 GRAPHICS 7:GRAPHICS 3:POK
E 553,45#6*(RE<2):POKE 53277,
3:GOSUB 1220:Y#Y+64#RE-VL/2
IB 1020 FOR A=1774 TO 1787:POKE A
,0:NEXT A
PB 1030 IF RE=1 AND PLL#0 THEN FOR
A=1780 TO 1787:POKE PLL#A,VL+128:NE
XT A:GOTO 1050
CD 1040 FOR A=0 TO NP:POKE PLL#A,
VL:NEXT A
DI 1050 P#M=255-NP#8#SI
LQ 1060 FOR A=0 TO 3:PY(A)=PMB#51
2#A#4+25#RE#A:NEXT A
RK 1070 FOR A=0 TO NP:FOR B=1 TO
VL:POKE P(A+B),B:PEEK(P(A+B)):N
EXT B:NEXT A
JL 1080 POKE 1788,PM+2#RE:Z#USR(1
636)
TT 1090 ? " ? "RETURN - PROGRAM
GLANY "1? "K" ZMIANA KOLORU
:?"? "M" ZMIANA WIELKOSCI
QA 1100 ? " ? "RE" ZMIANA ROZDZIELCZO
SCI Z "RE":NA "3:RE)
ED 1110 S=STICK(0):X#0
PF 1120 IF S<12 THEN X#X+SGN(8-S):S
#S+(S<8)*4+4:Y#Y+(X#X<0)*X<0-(
X#X>0)*X#YMIN
RN 1130 FOR A=0 TO NP:PX(A)=P(A)+
X:NEXT A
WI 1140 IF S<15 THEN Y#Y+SGN(13,5
-S):Y#Y+(Y<0)-(Y<25-VL)
BJ 1150 FOR A=0 TO NP:POKE PLY#A,
Y:POKE PLX#A,PK(A):NEXT A
WI 1160 IF PEEK(53775)=255 THEN 1
110
HY 1170 GOSUB 400:IF A#155 THEN G
050 1230:GOSUB 170:GOTO 900
KC 1180 IF A#ASC("M") THEN 1250
ZG 1190 IF A#ASC("K") THEN 1300
FN 1200 IF A#ASC("R") THEN RE#3-R
E:GOSUB 1230:GOTO 1010
NO 1210 GOTO 1110
SB 1220 PX(0)=128-4*(NP+1)*SI:FOR
A=0 TO NP:PX(A)=PX(0)+8#SI#A:
POKE PLX#A,PK(A):POKE 53256#A,
SI-1:NEXT A:RETURN
KC 1230 FOR A=0 TO 3:POKE PLY#A,0
:POKE PLY#A,0:NEXT A:RETURN
GI 1240 FOR A=704 TO 707:POKE A,C
:0:NEXT A:RETURN
YU 1250 ? "WIELKOSC = "SI:?"? "NAC
IENSIJ 1 2 LUB 4 DLA ZMIANY
WIELKOSCI:?"? "NACIENSIJ:RETURN
ABY W
JSC Z PROCEDURY ZMIANY WIELKOS
CI:?"?
ZL 1260 GOSUB 400:IF A#155 THEN 1
090
GZ 1270 IF A#49 OR A#50 OR A#52 T
HEN SI#A-48:GOSUB 1220
RE 1280 GOTO 1250
IC 1300 A#INT(CO/16):B#CO-A#16:?"
? "KOLOR="A":JASKRAWOSC = "B":
B
ER 1310 ? "POZIOME RUCHY MANIPULA
TORA - ZMIANA KOLORU, PIONOWE
E - ZMIANA JASKRAWOSCI:?"?
HF 1320 ? "RETURN" WYJSCIE Z PRO
CEDURY:?"?
IZ 1330 S=STICK(0):IF PEEK(53775)
<255 THEN 1380
XN 1340 IF S#1 THEN 1330
YI 1350 IF S#7 OR S#11 THEN A#A#S
GN(8-S):A#A-(A#15):B#A(0)*16
XD 1360 IF S#13 OR S#14 THEN B#B-
7

```

```

2KSGN(13.5-S):B=B-(B>14)*16+(B
<8)>16
QV 1370 CO=16*A+B:GOSUB 1240:GOTO
1390
ZT 1380 GOSUB 400:IF A=155 THEN 1
090
PF 1390 GOTO 1300
YC 1400 GOSUB 200:COLOR 1:FOR A=0
TO NP:K=XMIN+AKXD1&B:FOR Y=1
TO VL
JY 1410 B=PEEK(P(A)+Y):C=129:FOR
X1=0 TO 7:IF MI THEN FOR X1=0
TO 1
AQ 1420 IF B=C THEN B=B-C:GOSUB
1440
ZL 1430 C=C/2:NEXT X1:NEXT Y:NEXT
A:K=28&G:Y=10&G-1:GOSUB 200:G
OTO 1200
YZ 1440 PLOT X=X1&XD1,Y=MIN+Y-1:DR
AWO X=X1&XD1+XD,Y=MIN+Y-1:RETU
RN
SH 1450 GRAPHICS 0:FOR A=0 TO NP:
? :IF MI THEN ? "POCISK":GOTO
1470
WM 1460 ? "GRACZ ":A
QM 1470 FOR B=1 TO VL:PEEK(P(A)+
B):? :NEXT B:? CHR(126):NE
XS A
IS 1480 ? ? "NACISNIJ <RETURN>
ABY:WROCIĆ DO PROGRAMU":GOSUB
400:GOTO 900
AY 1500 ? ? "JAKI JEST NR TEGO G
RACZA, POCISKU LUB 1-SZEJ GRUP
Y GRACZY":TRAP 900:INPUT PN
BM 1510 ? ? "UMIEŚC CZYSTĄ TASMĘ
W MAGNETOFONIE - NACISNIJ P
RY, RECORD I RETURN"
AG 1520 OPEN #1,0,0,"C":FOR A=1
TO 64: ? #1,"R.":NEXT A
ZE 1530 FOR A=0 TO NP:LI=31400+(P
N+A)*100: ? #1:IF MI THEN 1600
JX 1540 ? #1:LI:"REMK* GRACZ
":A:"ROZDZIELCZOSC,DLUGOSC,WIE
LKOSC,KOLOR.":GOSUB LA
WT 1550 ? #1:LI:"DATA ":REJ:,"VL
":,"SI":GOSUB LA:GOTO 1560
EI 1560 FOR B=1 TO VL:IF B=1 OR B
=25 THEN ? #1: ? #1:LI:"DATA ":
:GOSUB LA:GOTO 1560
AX 1570 ? #1: ?
TT 1580 ? #1:PEEK(P(A)+B):NEXT B
NEXT A:CLOSE #1:TRAP 900:INPU
T A:GOTO 900
JY 1590 LI=LI+10:RETURN
TM 1600 LI=31000+(PN+A)*100: ? #1:
LI:"REMK* POCISK ":PN:"ROZDZIE
LCZOSC,DLUGOSC,WIELKOSC.":GOSU
B LA
JL 1620 ? #1:LI:"DATA ":REJ:,"VL
":,"SI":GOSUB LA:GOTO 1560
YQ 9000 FOR I=1536 TO 1706:READ A
POKE I,A:NEXT I:RETURN
OS 9010 DATA 162,3,169,244,6,248,
69,56,221,248,6,149,63,141,254
,6,106,141
DR 9020 DATA 255,6,142,253,6,24,1
69,0,109,253,6,24,109,252,6,13
3,204,133
FO 9030 DATA 206,109,240,6,133,20
3,173,254,6,133,205,109,248,6,
170,252,46,253
GN 9040 DATA 6,144,16,168,177,203
,145,205,169,0,145,203,136,202
,208,244,76,87
XO 9050 DATA 6,160,0,177,203,145,
205,163,0,145,203,200,202,208,
244,174,253,6
GW 9060 DATA 173,254,6,157,240,6,
189,236,6,240,48,133,203,24,13
0,141,253,6
JY 9070 DATA 109,235,6,133,204,24
,173,253,6,109,252,6,133,206,1
05,240,6,133
MC 9080 DATA 205,109,240,6,170,16
0,0,177,203,145,205,200,202,20
8,240,174,253,6
FQ 9090 DATA 169,0,157,236,6,202,
48,3,76,2,6,76,98,220,0,0,104,
169
QC 9100 DATA 7,162,6,160,0,32,92,
220,96

```

(K)olor — do zmiany koloru gracza. Poziomy ruch manipulatora zmienia kolor. Pionowy ruch zmienia jasrakość. Kody liczbowe koloru i jasrakości są wyświetlane po dokonaniu zmiany.

(R)ozdzielczość — do zmiany rozdzielczości.

(W)yświetlić: Wyprowadza dane obrazu gracza.

(P)oprawić: Powrót do sztalugi kreślenia. Jeśli wielkość lub rozdzielczość były zmieniane w czasie obserwacji, wtedy odpowiednio modyfikowana jest sztaluga. Jeśli ograniczenia wielkości zostały przekroczone, wtedy wielkość zredukowana jest do dwóch.

(Z)apamiętać: Przechowuje dane numeryczne obrazu na taśmie. Władczą taśmę do magnetofonu. Naciśnij RECORD i PLAY. Odpowiedź na pytanie o numer gracza. Jeśli to pierwszy gracz, odpowiedź zero. Jeśli poprzedni gracz był przechowany, wprowadź kolejny numer. Na przykład: jeśli pierwsze kreślenie miało szerokość dwóch graczy, będą to gracie zero i jeden. Dlatego wprowadź teraz numer dwa. Dane będą przesyłane na taśmę przy użyciu instrukcji PRINT.

(R)ozpocząć z nowym graczem: Używane do kasowania obrazu aktualnego gracza i rozpoczęcia kreślenia nowego.

D. Wykorzystanie zapisanych danych

Zapisane na taśmie dane mogą być wprowadzone do komputera przez instrukcję ENTER. Mogą być one połączone z rezydującym w pamięci programem. Instrukcje DATA będą miały numery zaczynające się od 31 000.

A. Jeśli program nie ma być łączony z rezydującym programem, wprowadź NEW i naciśnij <RETURN>.

B. Umieść taśmę w magnetofonie.

C. Naciśnij PLAY.

D. Wprowadź ENTER, C i naciśnij dwa razy <RETURN>.

E. Przed zakończeniem wprowadzania na ekranie wystąpi dwa razy READY.

F. Wprowadź (za pomocą LIST) to, co wprowadziłeś.

Efekt specjalne

Każda próba przemieszczania punktów w trybie graficznym o dużej rozdzielczości, pionowo lub ukośnie, z dużą szybkością, jest zwykle trudna, jeśli nie niemożliwa do wykonania w BASIC'u. Jest to dosyć trudne do uzyskania nawet w języku maszynowym.

Chcemy zademonstrować prostą technikę otrzymania skomplikowanych ruchów, dla uzyskania specjalnych efektów, wykonaną całkowicie w BASIC'u. Aby to zobaczyć, należy wprowadzić program nr 5 i uruchomić go. Składa się on z dwóch części.

Druga działa w nieskończonej pętli i w celu jej przerwania należy nacisnąć klawisz RESET.

Jak widać, program napisany jest całkowicie w BASIC'u i nie ma tu skomplikowanych obliczeń. Aby się dowiedzieć, jak to działa, proszę uruchomić program przez napisanie **GOTO 1200**.

Idea tej techniki była krótko opisana w poprzednim odcinku. Opis dotyczył używania maski pola gry do zwiększenia rozdzielczości gracza. W tym programie po prostu rozwinęto tę ideę przez użycie maski jako matrycy, która tworzy iluzję ruchu w pionie i po przekątnej w czasie, gdy pionowy pasek gracza przesuwany jest poziomo po ekranie.

Poszczegółone etapy uzyskiwania takich efektów są następujące:

- 1) Kreślenie maski poprzez wypełnienie ekranu lub wybranego obszaru kolorem pola gry; ustawienie koloru i jego szerokości do takich jak to.
- 2) Kreślenie matrycy w kolorze tła (aby była niewidoczna).
- 3) Tworzenie pionowego paska o odpowiedniej wysokości i szerokości za pomocą grafiki gracza—pocisk.
- 4) Ustawienie priorytetu pola gry nad graczem i przesuwaniu paska poziomo poza maskę.

Aby lepiej zobaczyć, jak to jest zrobione, prześledźmy pierwszą część programu:

linia 160 — wybranie trybu graficznego;

linia 790 — czyszczenie ekranu w procesie kreślenia poprzez ustawienie jego koloru na czarny;

linia 490 — kreślenie maski w kolorze 3 pola gry;

linie 500—520 — kreślenie matrycy dla wystrzału w kolorze 0 (tła);

linie 530—570 — kreślenie jabłka w kolorze 1 pola gry;

linie 680—730 — kreślenie matrycy dla rozbrzygnięcia jabłka w kolorze 0;

linie 740—770 — kreślenie karabinu w kolorze 2 pola gry;

linie 820—890 — przygotowanie grafiki gracza—pocisk;

linia 850 tworzy gracza nr 2 w postaci paska o szerokości 5 bitów rozciągającego się od góry do dołu ekranu,

linie 860—870 — tworzą gracza nr 1 jako pocisk karabinowy,

linia 830 ustala priorytet (za pomocą POKÉ 623,2), gracz nr 1 jest przed frontem maski, a gracz nr 2 jest poza maską,

linie 190 i 250 — ustawianie kolorów;

linie 260 do 320 — gracz nr 2 przesuwa się poziomo przez ekran i jego kolor zmienia się od białego do czerwonego (gdą przechodzi poza jabłkiem).

Ludwik PIELA

Tomasz MROWIEC

```

FK 1 REM *****
FJ 2 REM *
LU 3 REM * PROGRAM nr 5 *
FL 4 REM *
FO 5 REM *****
JI 10 N1=784:N2=862:N3=0:GOTO 150
      REM SKOK DO PROGRAMU GLOWNEGO
KT 20 REM OPOZNIENIE
JO 30 FOR D=1 TO DELAY2:NEXT D:RE
      TURN
NL 40 REM DZINIEK
MM 50 SOUND 0,17,10,10
LR 60 FOR D=1 TO 4:NEXT D
MW 70 SOUND 0,0,0,0:RETURN
ZT 80 REM OSCYLOSKOP
UY 90 FOR H=13 TO 200 STEP 2
DG 100 POKE 53250,H
GR 110 IF H=55 OR H=93 OR H=131 OR
      H=169 THEN GOSUB 50
KP 120 FOR D=1 TO 1:NEXT D
XI 130 NEXT H:GOTO 90
CO 140 REM USTAWIENIE PARAMETROW
UI 150 DELAY2=500:FLAG=0:G=3
DR 160 GRAPHICS 7:GOSUB 790:POKE
      709,10:POKE 752,1:POKE 657,10:
      "STRZAL W JABLKO"
ZW 170 POKE 656,1:POKE 657,12: "
      (za 14 sek.)"
SL 180 GOSUB 820:GOSUB 490
HR 190 POKE 709,68:POKE 709,6:POK
      E 712,0
AU 200 SOUND 0,30,10,14:FOR D=1 T
      O 25:NEXT D:SOUND=0,0,0,0
YH 210 REM STRZAL W JABLKO
BF 220 POKE 657,17:CHR$(199):CH
      R$(207):CHR$(212):CHR$(207):CH
      R$(215):CHR$(217):GOSUB 30
TA 230 ? CHR$(125):POKE 657,18: "
      "CEL":GOSUB 30
LS 240 ? CHR$(125):POKE 657,18: "
      CHR$(208):CHR$(193):CHR$(204):
      GOSUB 30
LB 250 POKE 706,12:POKE 705,79
TY 260 FOR L=1 TO 35:SOUND 0,L,4,
      15:NEXT L
RH 270 FOR V=15 TO 0 STEP -G
CF 280 POKE 705,64+V:SOUND 0,L,4,
      V
CU 290 NEXT V:POKE 705,0
GI 300 FOR H=60 TO 122 STEP G:POKE
      53250,H:NEXT H
GD 310 POKE 706,0:POKE 706,68
EE 320 FOR H=123 TO 200 STEP G:PO
      KE 53250,H:NEXT H
VF 330 POKE 706,0:POKE 53250,0:FO
      R D=1 TO DELAY2:NEXT D:IF FLAG
      THEN 430
XI 340 POKE 752,1:CHR$(125):POK
      E 657,6: "TO SRMO W MOLLNIEJSZ
      YM TEMPIE."

```

```

YB 350 FOR D=1 TO DELAY2:NEXT D:G
      =1
HP 360 POKE 706,68:FOR D=1 TO DEL
      AY2:NEXT D:CHR$(125)
DL 370 FOR H=200 TO 123 STEP -3:P
      OKE 53250,H:NEXT H
GB 380 POKE 708,68:POKE 706,12
FR 390 FOR H=122 TO 60 STEP -3:PO
      KE 53250,H:NEXT H
XI 400 FOR D=1 TO DELAY:NEXT D
KI 410 POKE 53250,250:FLAG=1:DELA
      Y2=800:GOTO 220
XM 420 REM USTAWIENIE PARAMETROW
      OSCYLOGRAMU
GO 430 POKE 53249,0:POKE 53250,0
      KE 53250,H:NEXT H
FT 440 POKE 709,6:CHR$(125):POK
      E 657,15: "OSCYLOSKOP"
UV 450 GOSUB 910
RG 460 FOR I=PMB+1 TO PMB+N2:POK
      E I,3:NEXT I
DK 470 POKE 706,14:CHR$(125):GO
      TO 90
QZ 480 REM RYSOWANIE JABLKA
TY 490 COLOR 3:FOR Y=0 TO 79:PLOT
      0,Y:DRAWTO 159,Y:NEXT Y
VK 500 COLOR 0:FOR Y=38 TO 42 STE
      P 0
LO 510 FOR Y=14 TO 72 STEP 2:PLOT
      0,Y:NEXT X:NEXT Y
MC 520 PLOT 74,38:PLOT 74,42
VL 530 COLOR 1:Y1=35:X2=82
LS 540 FOR X1=77 TO 74 STEP -1
LI 550 Y1=Y1+1:X2=X2+1
OT 560 PLOT X1,Y1:DRAWTO X2,Y1
GU 570 NEXT X1
UY 580 FOR Y1=40 TO 41
FO 590 PLOT 74,Y1:DRAWTO X2,Y1
UV 600 NEXT Y1:Y1=41
FE 610 FOR X1=75 TO 78
NO 620 Y1=Y1+1:X2=X2+1
MO 630 PLOT X1,Y1:DRAWTO X2,Y1
GR 640 NEXT X1
UX 650 PLOT 80,36:DRAWTO 80,34:PL
      OT 81,34
KY 660 PLOT 81,33:PLOT 82,33:PLOT
      82,32
IZ 670 COLOR 3:PLOT 79,36:PLOT 81
      35
KT 680 COLOR 0
YJ 690 PLOT 83,35:DRAWTO 120,0:PL
      OT 84,36:DRAWTO 140,0
TP 700 PLOT 85,37:DRAWTO 159,4:PL
      OT 86,38:DRAWTO 159,16
T1 710 PLOT 87,39:DRAWTO 159,28:P
      LOT 87,40:DRAWTO 159,40
PH 720 PLOT 87,41:DRAWTO 159,52:P
      LOT 86,42:DRAWTO 159,64:PLOT 8
      5,43:DRAWTO 159,76
OB 730 PLOT 84,44:DRAWTO 140,79:P
      LOT 83,45:DRAWTO 120,79

```

```

GO 740 POKE 709,0:COLOR 2:FOR Y=3
      7 TO 43
MB 750 PLOT 0,Y:DRAWTO 13,Y
MN 760 NEXT Y
ZU 770 PLOT 14,37:PLOT 14,43: ? CH
      R$(125):RETURN
OS 780 REM CZYŚCZENIE EKRANU
PR 790 FOR H=0 TO 4:POKE 708+H,0
      :POKE 710,H3
ZE 800 RETURN
EO 810 REM USTAWIENIE PARAMETROW
      P/M
HU 820 POKE 705,0:POKE 706,0:A=P
      E(106)+24:POKE 53249,0:POKE 5
      3250,0
BI 830 POKE 54279,A:PMB=256+A:POK
      E 53277,2:POKE 623,2:POKE 559,
      46
MV 840 FOR I=PMB+640 TO PMB+695:P
      OKE I,0:NEXT I
ZS 850 FOR I=PMB+1 TO PMB+N2:POK
      E I,31:NEXT I
HY 860 POKE 870,0:FOR I=PMB+634
      TO PMB+698:READ A:POKE I,A:NEX
      T I
HT 870 DATA 192,240,254,240,192
      110 880 POKE 53257,0:POKE 53258,0
      :POKE 53249,62
ZM 900 RETURN
AL 900 REM WYKRESLANIE OSCYLOGRAMU
      U
IA 910 TRAP 1000
TP 920 COLOR 3:FOR Y=0 TO 79:PLOT
      0,Y:DRAWTO 159,Y:NEXT Y
YU 930 COLOR 0:X=0
LZ 940 PLOT X,32:DRAWTO X,2
OM 950 X=X+12
ID 960 DRAWTO X,2:DRAWTO X,32
RA 970 X=X+12
PB 980 DRAWTO X,32:DRAWTO X,2
LR 990 GOTO 950
NO 1000 TRAP 40000:Y=59:HT=-15:SL
      =-3
OI 1010 FOR X=0 TO 114 STEP 10
XO 1020 PLOT X,Y
LJ 1030 FOR XX=0 TO 19 STEP 19
ML 1040 DRAWTO XXX+10,Y
MS 1050 DRAWTO XXX+13,Y+SL
MY 1060 DRAWTO XXX+14,Y+SL
OG 1070 DRAWTO XXX+14,Y+SL+HT
OU 1080 DRAWTO XXX+14,Y+SL
UY 1090 DRAWTO XXX+14,Y+SL
BX 1100 DRAWTO XXX+18,Y
XA 1110 SL=SL-HT:NEXT
      YL 1120 NEXT XX:HT=HT X:DRAWTO 159
      HT
CB 1200 N1=789:N2=892:N3=4:GOTO 1
      50

```

PROGRAM

AMSTRAD

25

3 punkty

Program wymaga wczytania współrzędnych trzech punktów na płaszczyźnie, po czym podaje komunikat o ich wzajemnym położeniu. Jeśli okaże się, że wszystkie leżą na jednej prostej, dodatek pojawi się tylko równanie w postaci $y=f(x)$. Metoda zastosowana w programie polega na porównaniu długości odcinków łączących punkty.

Program pozbowiany jest instrukcją skoków, lecz jego struktura nie dorównuje Pascalowi. Problem tu przedstawiony

nie daje się jednak uchwycić w Pascalu. Dlaczego? Spróbujcie odpowiedzi poszukać sami...

Krzysztof MAMCARZ

```

10 * ## 3 punkty * B6
20 DEFINT I
30 DEFSTR W
40 DEFREAL A-D,M,N,X,Y
50
60 DEF FN dlugosc(x1,x2,y1,y2)=ROUND(SQR
  ((x1-x2)^2+(y1-y2)^2),5)
70 DEF FN n(x1,x2,y1,y2)=(y2-y1)/(x2-x1)
80 DEF FN n(x1,y1,m)=y1-m*x1
90 DIM X(3)
100 I=1
110
120 HODE I
130 W$="xywzbykbcyc"
140 PRINT "PODAJ WSPOLRZEDNE PUNKTOW A,B
  ,C"

```

```

150 FOR I=1 TO 3
160 PRINT MID$(W$,48I-3,2):"=";
170 INPUT X(I)
180 PRINT MID$(W$,48I-1,2):"=";
190 INPUT Y(I)
200 NEXT
210 PRINT
220
230 AB=FN dlugosc(X(1),X(2),Y(1),Y(2))
240 BC=FN dlugosc(X(2),X(3),Y(2),Y(3))
250 AC=FN dlugosc(X(1),X(3),Y(1),Y(3))
260 IF AB=0 AND BC=0 AND AC=0 THEN PRINT
  "A=B=C:PRINT przez 1 punkt przechodzi
  dwa prostych"END
270 IF AB=0 OR BC=0 OR AC=0 THEN PRINT "
  przez dwa punkty przechodzi"PRINT"ZAMB
  E JAKO PROSTA"END
280 C=ROUND(ABS((AC-BC),4)
290 AB=ROUND(AB,4)
300 IF (AB=0) THEN ELSE PRINT "punkty A,
  B,C nie sa wspolnolineowe..."END
310 PRINT "punkty A,B,C sa wspolnolineowe"
320 IF X(1)=X(2) AND X(2)=X(3) THEN PRIN
  T "prosta jest||"END
330 IF Y(1)=Y(2) AND Y(2)=Y(3) THEN PRIN
  T "prosta jest||"END
340 X=X(X(2),Y(2),Y(1),Y(2))+FN FN(X
  (1),Y(1),m)
350 PRINT "prosta y=";
360 IF AC=0 AND BC=1 THEN PRINT ac||CHR(8
  )||"; ELSE IF ac=1 THEN PRINT "1"
370 IF n=0 THEN PRINT "+||" ELSE IF n<0
  THEN PRINT "-"
380 END

```

PROGRAM

ATARI

Interpolacja Lagrange'a

Zdarzają się sytuacje, gdy mając funkcję w postaci dyskretnej, tzn. zbioru punktów X_i i Y_i ($i = 1, 2, \dots, N$); trzeba uzyskać jej przybliżony wzór. Jedną z metod otrzymania tego wzoru jest utworzenie, a następnie przekształcenie układu równań, z którego wylizymy współczynniki wielomianu interpolującego funkcję daną tablicą. W poniższym przykładzie zawężono otrzymane wielomiany do stopnia pierwszego i drugiego, a więc do ich wyznaczenia potrzebne będą dwie lub trzy współrzędne punktów. Kolejność wprowadzania współrzędnych punktów jest dowolna przy czym jako pierwszy deklarujemy X a po nim Y . Jeżeli jeden z X -ów tablicy jest równy zero, wprowadzamy go w ostatniej kolejności, gdyż ze względu na sposób wyprowadzenia wzorów zadanie może okazać się niewykonalne.

Janusz W. JANIEC

```

50 REM *****
55 REM * INTERPOLACJA *
65 REM *
70 REM * LAGRANGE'A *
75 REM *
90 REM *****
100 OPEN #1:4,B,"K"
110 ? CHR$(125)
120 ? "Ile masz danych punkto
w w tablicy?"
150 ? GET #1,ST:PO=ST-49
155 IF PO<1 OR PO>2 THEN 110
160 ? CHR$(28):CHR$(28):"Wprowadz
zaj kolejno dane z tablicy"
170 ? "C Xn.Yn.Xn+1.Yn+1,...]
"
175 TRAP 410
180 ON PO GOTO 200,300
200 INPUT X1,Y1,X2,Y2
210 B=(Y2*X1-Y1*X2)/(X1-X2)
220 A=(Y1-B)/X1
230 ? CHR$(125)
240 POSITION 2,10: ? "Y=";A;"XX"

```

```

250 IF B<0 THEN ? CHR$(30);B:GOT
0 420
260 ? B:GOTO 420
300 INPUT X1,Y1,X2,Y2,X3,Y3
310 XH=X1-X2:XB=X2-X1:XC=X1-X2:YD
=XH*XB
320 XE=XD*XC:XF=X3-X2:XG=XB-X2
330 C=(Y3*XE+XF*(Y2*X1-Y1*(XD+XG
*(X1)))+(Y2-C*XG*(C-Y1))/XD
340 B=(Y2-C*XG*(C-Y1))/XD
360 ? CHR$(125)
380 POSITION 2,10: ? "Y=";A;"XX+
"
385 IF B<0 THEN ? CHR$(30);B:GO
TO 395
390 ? B;
395 ? "XX";IF C<0 THEN ? CHR$(
30):C:GOTO 420
400 ? C:GOTO 420
410 IF PEEK(195)=11 THEN ? "M
IANGOWNIK JEST=0, TRZEBA ZMIENIC'
X'"
420 POSITION 2,15: ? "C Wcisnij d
owolny klawisz"
430 ? GET #1,P:GOTO 110
440 REM *****

```

* SUMA KONTROLNA / ETYKIETA *

LK	501	QW	551	CC	601	QX	65
VI	701	QY	751	LN	801	PA	100
AU	1101	RL	1201	FT	1501	HP	155
VT	1601	PI	1701	MI	1751	SK	180
JM	2001	UB	2101	BG	2201	AZ	230
QE	2401	FE	2501	PA	2601	RY	300
GR	3101	FH	3201	KP	3301	AY	340
HP	3501	BG	3601	DJ	3801	EZ	385
UG	3901	CB	3951	OX	4001	ST	410
MN	4201	RN	4301	PN	4401		

PROGRAM

ATARI

Disassembler

Analizowanie programów napisanych w kodzie maszynowym jest czynnością bardzo żmudną i pracochłonną, lecz niezbędną szczególnie wtedy, gdy chcemy dobrze poznać algorytm działania programu. Najwięcej pracy w takiej sytuacji pochłania „przejście” z kodu maszynowego programu, zwanego również kodem wynikowym, na jego kod źródłowy, który jest czytelny dla programisty. Kod źródłowy składa się z mnemonicicznych nazw rozkazów mikroprocesora i przyporządkowanych im wartości argumentów (operandów), a jego uzyskiwanie z kodu wynikowego nazywa się disassemblacją. Komputer może w tej operacji z powodzeniem zastąpić programistą — wystarczy mieć odpowiedni program zwany disassemblerem.

Disassemblerem jest prezentowany program o nazwie DISASM. Wykonuje on disassemblację obszaru pamięci, którego początek i koniec określa użytkownik programu przez podanie odpowiednich adresów. Program do disassemblacji musi być najpierw wprowadzony do pamięci komputera. Program DISASM wczytuje do pamięci komputera programy, które mają być disassembledowane, zarówno z dysku, jak i z taśmy. Wyniki disassemblacji wyprowadzane są na monitor i jednocześnie na drukarkę, jeśli jest podłączona do komputera. Mają one następującą postać: adres pamięci, w którym znajduje się rozkaz, nazwa mnemoniciczna rozkazu i jego argument w postaci liczby dziesiętnej, kod dziesiętnej rozkazu i argumentu oraz kod heksadecymalny rozkazu i argumentu. Adres pamięci na monitor wyprowadzany jest w postaci dziesiętnej, a na drukarkę dodatkowo w postaci liczby heksadecymalnej.

Wszystkie rozkazy skoków w programie disassembledowanym są wyróżnione przez wydrukowanie dodatkowego znaku przed ich nazwą mnemoniczną. Znakiem tym jest

— gwiazdka, gdy rozkaz skoku jest skokiem bezwzględnym (rozkaży JMP i JSR);

— plus, gdy rozkaz skoku jest skokiem względnym do „przodu”, to znaczy, gdy skok wykonywany jest do adresu większego, niż adres, w którym znajduje się rozkaz skoku;

— minus, gdy rozkaz skoku jest skokiem względnym do „tyłu”, to znaczy, gdy skok wykonywany jest do adresu mniejszego niż adres, w którym znajduje się rozkaz skoku.

Rozkazami skoku względnego są rozkaży BPL, BMI, BVC, BCC, BCS, BNE, BEQ.

Miejsca, do których przekazywane jest sterowanie w disasembliowanym programie po wykonaniu rozkazu skoku względnego, podawane są w postaci adresów bezwzględnych obliczonych przez dodanie (gdy skok do „przodu”) lub odjęcie (gdy skok do „tyłu”) wartości adresu, w którym znajduje się rozkaz skoku i wartości jego argumentu. Takie rozwiązanie bardzo ułatwia analizowanie uzyskanego w wyniku disasemblacji kodu źródłowego.

Dla tych czytelników, którzy wolałoby posługiwać się adresami pamięci w postaci liczby heksadecymalnej, lub będą chcieli wprowadzać wyniki, inaczej niż to zrobiono w programie DISASM, podaje dokładną zawartość 70-bajtowej zmiennej tekstowej o nazwie AS. Zawiera ona wynik disasemblacji danego rozkazu, który następnie jest wyprowadzany na monitor i drukarkę. Poszczególne bajty tej zmiennej zawierają:

AS(15) — adres pamięci, w którym znajduje się disasembliowany rozkaz w postaci liczby dziesiętnej;

AS(7,11) — ten sam adres w postaci liczby heksadecymalnej;

AS(13,24) — nazwa mnemoniczna rozkazu oraz jego argument w postaci liczby dziesiętnej;

AS(29,39) — kod dziesiętnej rozkazu i argumentu,

AS(46,53) — kod heksadecymalnej rozkazu i argumentu.

Pozostałe bajty tej zmiennej zawierają spacje.

H. KRASUSKI

```
1 REM *****
2 REM
3 REM DISASSEMBLER
4 REM
5 REM *****
10 ? CHR$(125) ? "Proszę chwilke
poczekac, ..."
11 DIM SPEC$(16), CIO$(6) : CIO$(1,
1) = CHR$(104) : CIO$(2,2) = CHR$(162)
: CIO$(3,3) = CHR$(32)
12 CIO$(4,5) = "LV" : CIO$(6,6) = CHR$(
228) : CBUF=1000
14 J0=0 : J1=1 : J2=2 : J3=3 : J4=4
15 J18=9*J2
16 J9=900 : DRNAG=1100
25 DIM ROZ$(4), A$(70), C$(969), HE
X$(16), PK$(76)
27 RESTORE 30
28 FOR I=1 TO 75 : READ D : PK$(I)=C
HR$(D) : NEXT I
30 DATA 169,0,133,18,133,19,133,
20,104,104,133,204,104,133,203,1
60,0,177,203,153,206,0,200,192,4
,208,246
32 DATA 165,207,201,63,208,1,96,
24,163,203,185,4,133,203,144,2,2
30,204,160,0,177,203,197,205,240
,19,200
34 DATA 192,13,208,245,24,165,20
3,105,13,133,203,144,2,230,204,2
,4,144,199,152,133,18,96
40 RESTORE 45
42 FOR I=1 TO 969 : READ D : C$(I)=C
HR$(D) : NEXT I
45 DATA 32,65,60,67,2,105,101,11
7,2,109,125,121,2,2,97,113,2,32,
65,70,60,2,41,37,53,2,45,61,57,2
,2,33,49
47 DATA 2,32,65,83,76,10,2,6,2,22,
4,14,30,2,2,2,2,2,2,43,66,67,67,
2,2,2,2,2,2,2,2,144,2,2,2,2,43,6
6,67,63
49 DATA 2,2,2,2,2,2,2,2,2,2,176,2,
2,2,43,66,69,81,2,2,2,2,2,2,2,2,
2,240,2,2,2,32,66,73,84,2,2,36,2
,2,44,2
51 DATA 2,2,2,2,2,2,43,66,77,73,
2,2,2,2,2,2,2,2,48,2,2,2,43,66
,78,69,2,2,2,2,2,2,2,2,2,208,2,2
,2,43
53 DATA 66,80,76,2,2,2,2,2,2,2,2,
2,16,2,2,2,32,66,82,75,2,2,2,2,2
,2,2,2,2,0,2,2,2,2,2,43,66,86,67,2,
2,2,2,2
55 DATA 2,2,2,2,80,2,2,2,2,43,66,8
6,83,2,2,2,2,2,2,2,2,112,2,2,2,
32,67,76,67,2,2,2,2,2,2,2,2,2,24,
2,2,2,2
57 DATA 32,67,76,69,2,2,2,2,2,2,2,
2,2,2,16,2,2,2,32,67,76,73,2,2,2
,2,2,2,2,2,2,80,2,2,2,2,32,67,76,
86,2,2,2
59 DATA 2,2,2,2,2,2,184,2,2,2,2,32
,67,77,80,2,201,197,213,2,205,22
,1,17,2,2,193,209,2,32,67,80,88,2
,224,228
61 DATA 2,2,236,2,2,2,2,2,2,2,32,
68,69,67,2,2,198,214,2,206,222,
2,2,2,2,2,2,32,68,69,88,2,2,2,2,
```

```
2,2,2,2
63 DATA 202,2,2,2,2,32,68,69,89,
2,2,2,2,2,2,2,2,136,2,2,2,2,32,6
9,79,82,2,73,69,85,2,77,93,89,2,
2,65,81,2
65 DATA 32,73,78,67,2,2,230,246,
2,238,254,2,2,2,2,2,2,32,73,78,8
8,2,2,2,2,2,2,2,2,232,2,2,2,2,32
,73,78,89
67 DATA 2,2,2,2,2,2,2,2,200,2,2,
2,2,42,74,77,80,2,2,2,2,2,2,2,2,2
,2,2,2,2,2,108,42,74,83,82,2,2,2,2
,2,32,2,2
69 DATA 2,2,2,2,2,32,76,68,65,2,
169,165,181,2,173,189,185,2,2,16
1,177,2,32,76,68,80,2,162,166,2,2,
182,174,2
71 DATA 190,2,2,2,2,2,2,2,200,2,2,
2,2,42,74,77,80,2,2,2,2,2,2,2,2,2
,2,2,2,2,2,108,42,74,83,82,2,2,2,2
,2,32,2,2
73 DATA 2,2,2,2,32,76,68,65,2,
169,165,181,2,173,189,185,2,2,16
1,177,2,32,76,68,80,2,162,166,2,2,
182,174,2
75 DATA 2,2,2,2,2,2,2,2,200,2,2,
2,2,42,74,77,80,2,2,2,2,2,2,2,2,2
,2,2,2,2,2,108,42,74,83,82,2,2,2,2
,2,32,2,2
77 DATA 184,2,2,2,32,80,76,80,2,2,
2,2,2,2,2,234,2,2,2,32,79,82,2,2,
5,2,9,5,21,2,13,29,25,2,2,2,1,17,2
,32,80,72
79 DATA 65,2,2,2,2,2,2,2,2,72,2,2,
2,2,32,80,72,90,2,2,2,2,2,2,2,2,2,
2,8,2,2,2,2,32,80,76,65,2,2,2,2,2,
2,2,2,2
77 DATA 184,2,2,2,2,32,80,76,80,2,2,
2,2,2,2,2,2,2,2,40,2,2,2,32,82,
79,76,42,2,38,54,2,46,62,2,2,2,2,
2,2,2,32
83 DATA 82,79,82,106,2,102,118,73
,110,126,2,2,2,2,2,32,82,84,107
,2,2,2,2,2,2,2,2,64,2,2,2,2,32,8
2,84,83
81 DATA 2,2,2,2,2,2,2,2,96,2,2,2,
2,32,83,66,67,2,233,229,245,2,2,
37,253,249,2,2,225,241,2,32,83,6
9,67,2,2
83 DATA 2,2,2,2,2,2,2,2,8,2,2,2,2,32
,83,69,68,2,2,2,2,2,2,2,2,248,2,2,
2,2,2,32,83,69,73,2,2,2,2,2,2,2,
2,120,2,2
85 DATA 2,2,32,83,84,65,2,2,133,
149,2,141,157,153,2,2,129,145,2,2,
32,83,84,88,2,2,134,2,150,142,2,2,
2,2,2,2,2
87 DATA 2,32,83,84,89,2,2,132,14
2,2,140,2,2,2,2,2,2,32,84,65,8
8,2,2,2,2,2,2,2,2,170,2,2,2,2,32,
84,65,89
89 DATA 2,2,2,2,2,2,2,2,168,2,2,
2,2,32,84,83,88,2,2,2,2,2,2,2,2,
186,2,2,2,2,32,84,88,65,2,2,2,2,2,
2,2,2,2
90 DATA 138,2,2,2,2,32,84,80,83,
2,2,2,2,2,2,2,194,2,2,2,2,32,8
4,89,65,2,2,2,2,2,2,2,2,152,2,2,
2,2,32
91 DATA 67,80,89,2,192,196,2,2,2
04,2,2,2,2,2,2,32,63,63,63,2,2,
2,2,2,2,2,2,2,2,2,2,2
92 GOSUB CBUF ? "Adres poczatkowy
w obszarze" : INPUT AP : AP=AP+LN :
J3 ? "Adres koncowy obszaru" :
INPUT FK
```

```
94 POKE 82,J0 : HEX$="0123456789AB
CDEF" : DRU=J0 ? " ?
95 ? "adres r-z w jez. "? "dzi
es.assemblera kod dzies. kod h
eks."
96 ? "-----"
97 TRAP 99 : LPRINT : DRU=J1 : GOSUB
DRNAG
99 TRAP 40000 : BRKF=J0
100 BYTE=PEEK(AP) : POKE 205, BYTE :
IF BYTE=2 THEN ROZ$=" ???":Z=9 : G
OTO 400
110 X=USR(ADR(MK$),ADR(C$)) : Z=PE
EK(18)+J1 : FOR A=J1 TO J2 : Z=ROZ$
(A) : CHR$(PEEK(A+205)) : NEXT A : G
OTO 400
130 B=INT(A/16)
135 B=J1+A : INT(A/16) * 16
137 ROZ$(C,C)=HEX$(B,B) : RETURN
400 IF ROZ$="" BRK" AND AP CAK THE
N AP=AP+J1 : BRKF=J1 : GOTO 100
401 FOR R=J1 TO J2 STEP 10 : A$(R) =
" " : NEXT A
402 IF ROZ$="" ??? THEN Z=9
405 A$(J3+J2-INT(CLOG(AP)))=STR$(
AP) : A$(13,15)=ROZ$
410 A=AP : C=J2-INT(GOSUB 395)
420 FOR C=J3 TO J1 STEP -J1 : GOSUB
B : 390 : NEXT C
430 A$(7,11)=ROZ$
435 IF BYTE=J3 THEN BYTE=44
438 IF BYTE=J0 THEN A$(31,31)="0
" : GOTO 450
440 A$(31-INT(CLOG(BYTE)))=STR$(
BYTE)
450 A=C : J2=GOSUB 395
460 C=1 : GOSUB 390 : A$(46,47)=ROZ$(
J1, J2)
470 GOTO 460+Z*30
490 A$(J18, J18) = "A" : AP=AP+J1 : FL=
J3 : GOTO J9
520 A$(19) = STR$(PEEK(AP+J1)) : A$(
J18, J18) = "A" : AP=AP+J2 : FL=J1 : GOTO
J9
550 A$(J18) = STR$(PEEK(AP+J1)) : AP
=AP+J2 : FL=J1 : GOTO J9
580 A$(J18) = STR$(PEEK(AP+J1)) : A$(
21, 22) = "X" : AP=AP+J2 : FL=J1 : GOTO
J9
610 A$(J18) = STR$(PEEK(AP+J1)) : A$(
21, 22) = "Y" : AP=AP+J2 : FL=J1 : GOTO
J9
640 A$(J18) = STR$(PEEK(AP+J1)) + PEE
K(AP+J2)*256 : AP=AP+J3 : FL=J2 : GOT
O J9
670 A$(J18) = STR$(PEEK(AP+J1)) + PEE
K(AP+J2)*256 : A$(23, 24) = "X" : AP=
AP+J3 : FL=J2 : GOTO J9
700 A$(J18) = STR$(PEEK(AP+J1)) + PEE
K(AP+J2)*256 : A$(23, 24) = "Y" : AP=
AP+J3 : FL=J2 : GOTO J9
730 AP=AP+J1 : FL=J3 : GOTO J9
760 AP=PEEK(AP+J1) : IF A=127 THEN
AP=A-256 : A$(13,13)=CHR$(45)
770 A$(J18) = STR$(A+AP) : AP=AP+
J2 : FL=J1 : GOTO J9
790 A$(J18, J18) = "C" : A$(J18+J1) = S
TR$(PEEK(AP+J1)) : A$(22) = "X" : AP
```